

**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky a mezioborových inženýrských studií

**Diplomová práce**

2010

Bc. Vladislav Chechelev

---

# **TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: Informační technologie

Využití akcelerometrů pro měření parametrů 3D křivek

Using accelerometers for measuring 3D curve parameters

## **Diplomová práce**

Autor: Bc. Vladislav Chechelev

Vedoucí práce: Doc. Ing. Petr Tůma, CSc.

Konzultant: Ing. Tomáš Martinec, Ph.D.

Rozsah práce a příloh: 50

Počet obrázků: 36



## **Prohlášení**

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít svou diplomovou práci či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 21. 5. 2010

Podpis:

## **Poděkování**

Poděkování patří vedoucímu mé práce Doc. Ing. Petru Tůmovi CSc za metodické rady a konzultace při zpracování diplomové práce.

## **Abstrakt**

Práce se zabývá problematikou využití akcelerometrů při měření 3D křivek. První část práce seznamuje čtenáře s teorií, navrženou pro měření křivek pomocí akcelerometrů, metodikou víceosého měření úhlů a způsoby modelování křivek. Ve druhé části jsou popsány zkonstruované přípravky, systém pro sběr a zpracování dat, hardware navržených měřících modulů a převodníku USB/RS485. Třetí část obsahuje popis software pro vizualizaci měřených dat včetně navrženého postupu pro práci s 3D modely.

**Klíčová slova:** akcelerometry, MMA7361, měření úhlů, modelování křivek, PIC18LF1220, PIC18F2550, 3D modely, USB/RS485

## **Abstract**

This thesis deals with problems of using accelerometers for measuring 3D curves. First part introduces to theory, designed for measuring curves using accelerometers, multi-axis angle measuring and methods for modelling curves. Second part describes designed tools and hardware for measuring modules and USB/RS485 converter. Third part includes software description for measured data visualization and description of method designed for making 3D models.

**Keywords:** accelerometers, MMA7361, angle measuring, curve modelling, PIC18LF1220, PIC18F2550, 3D models, USB/RS485

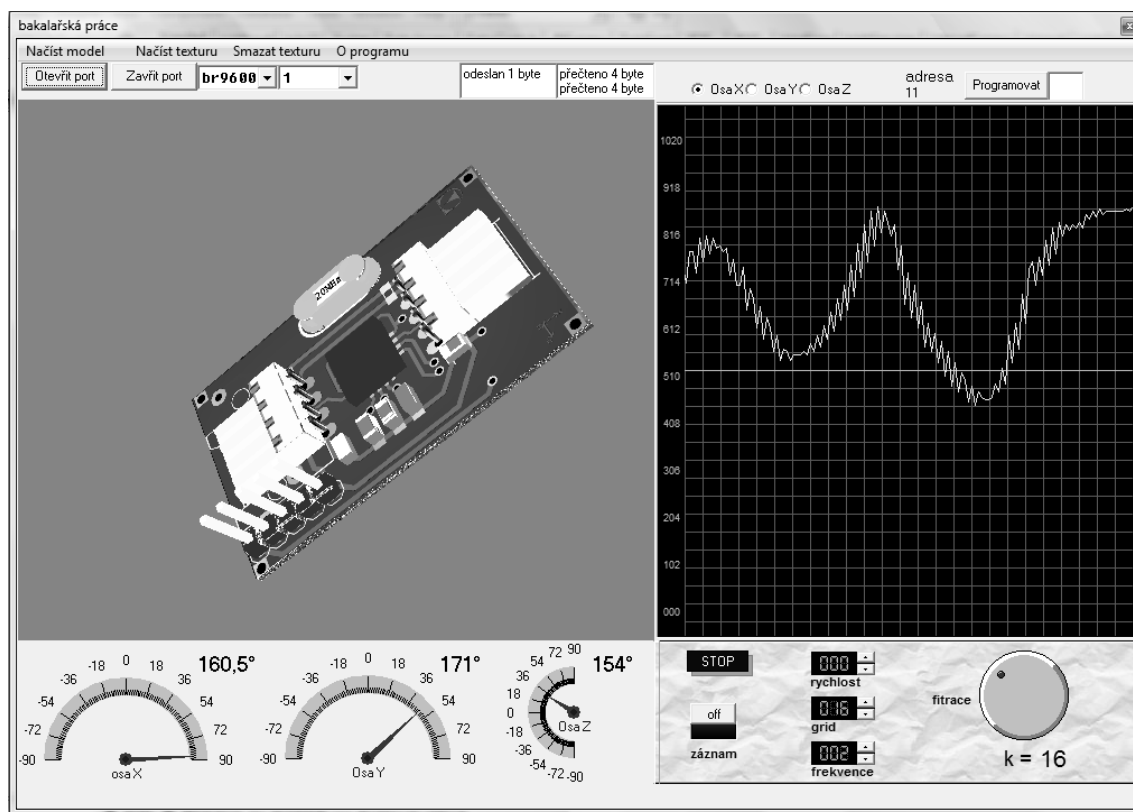
# Obsah

<b><u>1</u></b>	<b>Teorie .....</b>	<b>7</b>
<b><u>1.1</u></b>	Výběr senzoru .....	7
<b><u>1.2</u></b>	Metodika měření úhlů pomocí akcelerometrů .....	9
<b><u>1.3</u></b>	Možnosti spojování snímačů .....	13
<b><u>1.4</u></b>	Způsoby modelování křivek .....	14
<b><u>1.5</u></b>	Aproximace Bezierovou kubikou .....	15
<b><u>1.6</u></b>	Interpolace Catmull-Rom splajny .....	19
<b><u>1.7</u></b>	Porovnání metod modelování křivek .....	20
<b><u>1.8</u></b>	Měření křivek kloubově spojenými senzory .....	22
<b><u>2</u></b>	<b>Hardware .....</b>	<b>24</b>
<b><u>2.1</u></b>	Ideové schéma .....	24
<b><u>2.2</u></b>	Hardware modulů s akcelerometry .....	25
<b><u>2.3</u></b>	Firmware modulů s akcelerometry .....	27
<b><u>2.4</u></b>	Hardware převodníku USB/RS-485 .....	28
<b><u>2.5</u></b>	Firmware převodníku USB/RS-485 .....	30
<b><u>3</u></b>	<b>Software .....</b>	<b>32</b>
<b><u>3.1</u></b>	Metody zobrazování 3D modelů .....	32
<b><u>3.2</u></b>	Popis počítačových aplikací .....	37
<b><u>4</u></b>	<b>Závěr .....</b>	<b>41</b>
<b><u>5</u></b>	<b>Literatura .....</b>	<b>42</b>
<b><u>6</u></b>	<b>Obsah přiloženého CD .....</b>	<b>43</b>
<b><u>7</u></b>	<b>Přílohy .....</b>	<b>44</b>

# 1 Teorie

## 1.1 Výběr senzoru

Daná práce navazuje na práci bakalářskou, ve které jsem se zabýval problematikou měření odchylky od svislice pomocí akcelerometrů (hlavní okno programu lze vidět na obrázku č.1).

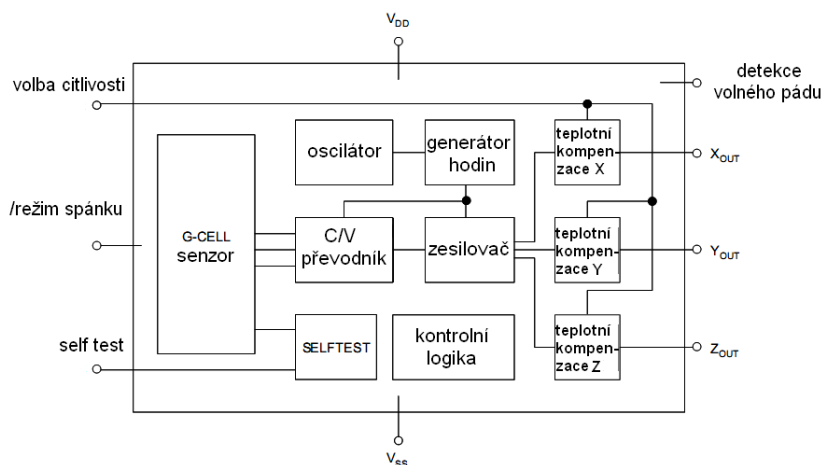


Obr. 1: Hlavní okno aplikace pro měření odchylky od svislice.

V bakalářské práci byly použity akcelerometry MMA7261, tyto snímače jsou analogové a pro získávání dat se musely použít dražší mikrokontrolery s analogově digitálním převodníkem. Proto tentokrát při výběru vhodných senzorů zrychlení volba padla nejprve na digitální akcelerometry MMA7455L s rozhraním I2C/SPI. Pokusné zapojení sestavené na nepájivém poli (viz příloha, str. 48) však ukázalo, že získávaných 62,5 vzorků za sekundu nestačilo pro efektivní filtraci signálu, nutnou pro zvětšení přesnosti senzoru na méně než jeden stupeň. Po srovnání vlastností vývojových

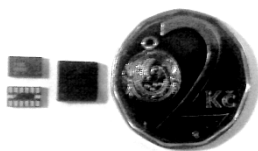


kitů pro akcelerometry MMA7455L, MMA7361 a MMA7660 na veletrhu Embedded World a konzultacích s technickou podporou firmy Freescale bylo rozhodnuto použít akcelerometr MMA7361L. Snímání hodnot z jeho analogových výstupů je omezeno pouze rychlosti A/D převodníku mikrokontroleru a vnitřní vzorkovací frekvencí akcelerometru (11kHz) (viz blokové schéma na obrázku č. 2).



Obr. 2: Blokové schéma akcelerometru MMA7361L.

MMA7361 je velmi levný tříosý kapacitní akcelerometr, oproti MMA7261 je ve skoro o polovinu menším pouzdře LGA14, má větší teplotní stabilitu a disponuje vylepšenou mikromechanickou strukturou, díky které je odolnější proti pádům a nárazům.



Obr. 3: Akcelerometry MMA7261 a MMA7361 v pouzdrech QFN16 a LGA14 3x5mm.

## 1.2 Metodika měření úhlů pomocí akcelerometrů

Akcelerometr MMA7361L je analogový a velikost napětí na jeho výstupech je dána vztahem:

$$V_{\text{out}} = V_{\text{off}} + S \cdot \sin \varphi \quad (1)$$

kde:

$V_{\text{out}}$  – napětí na výstupu akcelerometru [V]

$V_{\text{off}}$  – napětí na výstupech akcelerometru při 0g [V]

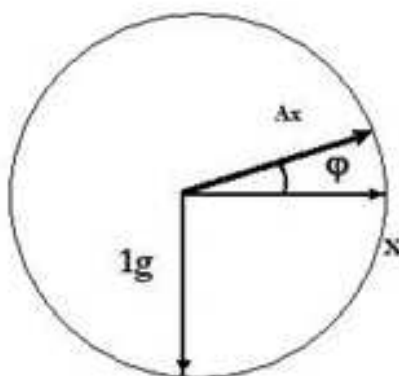
$\varphi$  – úhel, který daná osa svírá se směrem působení zemské gravitace [°]

$S$  – citlivost [ $\Delta V / \Delta g$ ]

Pro použitý rozsah 1,5g je hodnota citlivosti 800 mV/g.

Budeme-li měřit úhel natočení akcelerometru vůči vektoru zemské gravitace pomocí jedné osy, pak ze schématu obrázku č. 4 a vztahu pro výstupní napětí (1) lze odvodit vztah pro jeho velikost:

$$\varphi = \sin^{-1} \left( \frac{(V_{\text{out}} - V_{\text{off}})}{S} \right) \quad [^\circ] \quad (2)$$

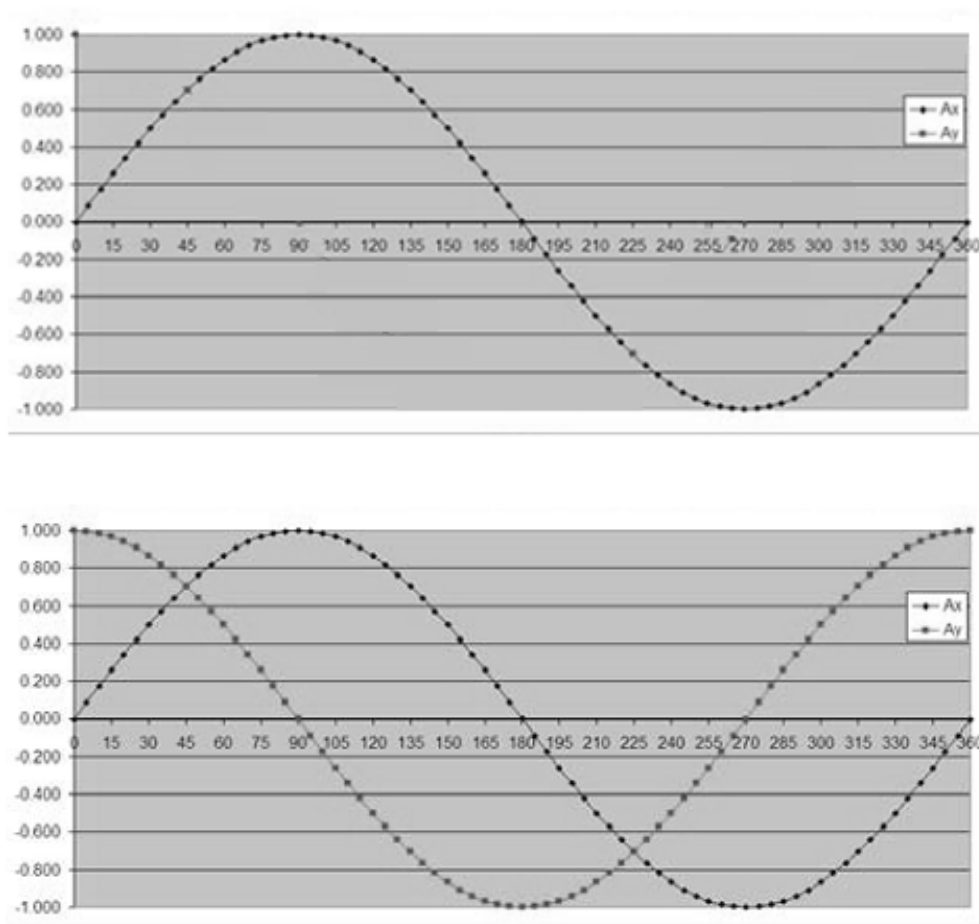


Obr. 4: Schéma odvození vztahu pro výpočet úhlu natočení pomocí jedné osy senzoru.

Část  $\frac{(V_{out} - V_{off})}{S}$  vyjadřuje zrychlení, budeme ho značit písmenem A (akceleraace) a můžeme psát

$$\varphi = \sin^{-1} A \text{ [}^\circ\text{]} \quad (3)$$

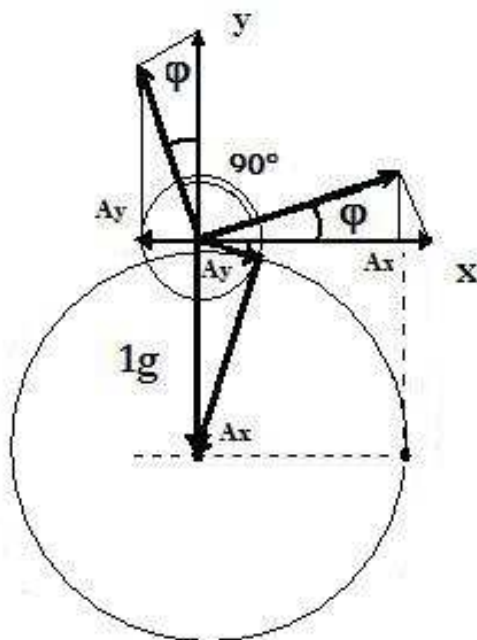
Nevýhodou daného způsobu měření je změna rozlišovací schopnosti senzoru v závislosti na jeho natočení. V grafických aplikacích daný problém způsoboval změnu plynulosti pohybu objektu. Jev lze pozorovat na grafu níže.



Obr. 5: Porovnání grafů závislosti zrychlení na natočení vůči zemi (a), (b): pomocí jedné osy (nahore) a dvou kolmých os (dole), z nichž je patrná změna rozlišovací schopnosti senzoru v závislosti na úhlech jeho natočení.

Cestou řešení je zahrnutí do výpočtu úhlu kombinace několika os.

Použijeme-li například navzájem kolmé osy **x** a **y**, zjistíme, že citlivost v ose **x** je nejmenší právě tehdy, když je osa **y** v maximu své citlivosti a opačně.



Obr. 6: Schéma odvození vztahu pro výpočet úhlu natočení kombinací dvou kolmých os senzoru.

Ze vztahu (3) a schématu na obrázku č. 6 odvodíme vztahy pro zrychlení v osách **x** a **y**:

$$Ax = \sin \varphi \quad [\text{g}] \quad (4)$$

$$Ay = \cos \varphi \quad [\text{g}] \quad (5)$$

Vydělíme-li je navzájem, pak dostaneme vztah pro výpočet úhlu užitím kombinovaného zrychlení:

$$\frac{Ax}{Ay} = \tan \varphi \quad (6)$$

$$\varphi = \tan^{-1} \frac{Ax}{Ay} \quad (7)$$

Ze schématu na obr. 5 je zřejmé, že součet zrychlení v x a y směrech se musí rovnat  $1g$ :

$$A = \sqrt{Ax^2 + Ay^2} = 1g \quad (8)$$

Ve tříosém akcelerometru je vektor gravitačního zrychlení rozkládán do třech složek:

$$A = \sqrt{Ax^2 + Ay^2 + Az^2} = 1g \quad (9)$$

Proto zcela analogicky byly odvozeny vztahy pro výpočet úhlů natočení užitím třech navzájem kolmých os:

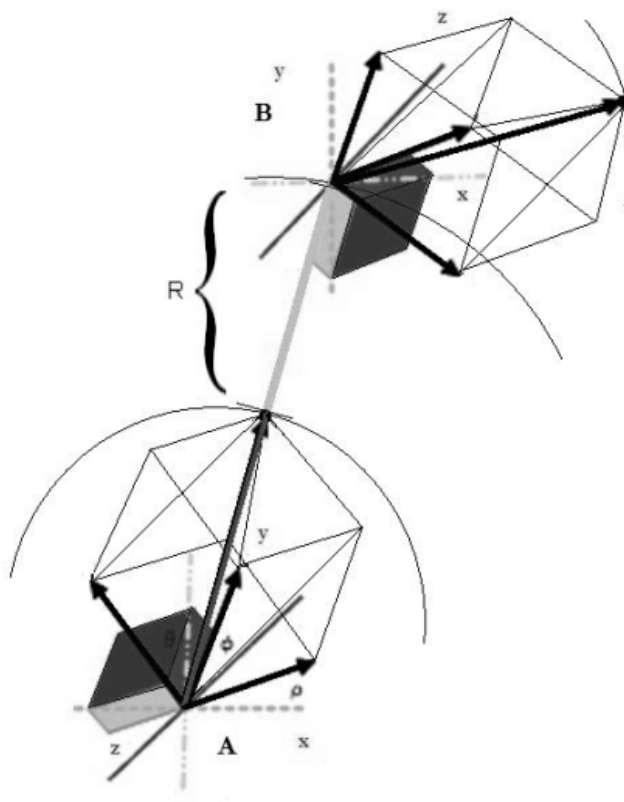
$$\varphi = \tan^{-1} \left( \frac{Ax}{\sqrt{Ay^2 + Az^2}} \right) \quad (10)$$

$$\phi = \tan^{-1} \left( \frac{Ay}{\sqrt{Ax^2 + Az^2}} \right)$$

$$\theta = \tan^{-1} \left( \frac{\sqrt{Ax^2 + Ay^2}}{Az} \right)$$

### 1.3 Možnosti spojování snímačů

Uvažujme nejjednodušší soustavu tuhých těles danou dvěma body ve 3D prostoru. Akcelerometr, umístěný v jednom z bodů nám poskytuje tři vektory, určující úhel jeho natočení vůči vektoru zemské gravitace. Pro zjištění polohy druhého senzoru, závislého na prostorové orientaci prvního, musíme mezi danými body definovat určitou vzdálenost, v modelu soustavy tuhých těles realizovanou vazbou: kloubovou nebo ohebnou. V rámci práce budou popsány způsoby měření a modelování obou typů spojení.



*Obr. 7: Znázornění podmínky pro zjištění polohy bodu  $B$  pomocí úhlů natočení senzoru v bodě  $A$ .*

Rekurzivní aplikací daného algoritmu lze postupně zjistit tvar celé množiny orientovaných bodů měřené křivky.

### 1.3 Způsoby modelování křivek

V zásadě existují dva druhy spojení soustav tuhých těles: kloubové a ohebné. Největším problémem při modelování soustav daných kloubově spojenými senzory je velký počet měřících modulů potřebných na jednotku délky modelovaného úseku. Ohebné spojování senzorů tento problém částečně řeší. Nesmíme zde pouze dopouštět, aby vznikající funkce křivky měla mezi každými dvěma senzory více než jeden extrém. Proto při rozmísťování měřících modulů, stejně jako při volbě nejvhodnějšího typu nahrazující křivky, je třeba vycházet v první řadě z mechanických

vlastností materiálu vazby modelovaného objektu a nastává problém výběru správné aproximační či interpolační metody, která by dávala dostatečně uspokojivé výsledky pro daný typ mechanické vazby. Existuje velké množství metod dovolujících modelování průběhu křivek na základě zadávání řídicích bodů. Obecně tyto metody lze rozdělit do dvou velkých skupin: interpolační a aproximační. Při interpolaci křivka prochází všemi řídicími body, zde připadaly v úvahu metoda interpolace Lagrangeovou, Fergusonovou křivkou, Catmull-Rom splajny. Při aproximaci je generovaná křivka řídicími body pouze vhodně řízena a nemusí jimi všemi procházet. Zde jsme volili mezi modifikacemi Bezierových křivek. Naprosto nepoužitelná byla Lagrangeova interpolační křivka, kvůli příliš prudkým obloukům a také proto, že její interpolační body musely být seřazeny postupně podle souřadnice  $x$ . Problematické bylo i použití Catmull-Rom splajnů, především proto, že výsledná křivka neležela v konvexní obálce řídicích bodů a neprocházela počátečním ani koncovým bodem. Podíváme-li se na fotografii na následující stránce, kde je zobrazen přípravek zkonstruovaný pro účely měření a modelování křivek ve 3D, uvidíme, že byly použity tři měřicí moduly. Proto při interpolaci byly použity tři úseky Catmull-Rom splajnů a při aproximaci stačila racionální Bezierova kubika. Oba vybrané typy křivek budou popsány v následujících kapitolách.



*Obr. 8: Fotografie přípravku pro měření parametrů křivek.*

## **1.5 Aproximace Bezierovou kubikou**

Při aproximaci modelu křivky Bezierovou kubikou bylo využito vlastností Bernsteinových polynomů, jejichž součet nepřesahuje 1:



$$\sum_{i=0}^n B_i^n(t) = 1$$

Bezierova křivka tak leží vždy v konvexní obálce svého řídícího polygonu, který lze snadno modelovat, pokud na každé jeho úsečce bude umístěn alespoň jeden senzor.

Obecně je Bezierova křivka definována následujícím vztahem:

$$P(t) = \sum_{i=0}^n P_i B_i \quad (11)$$

Rozepsáním vztahu (11) pro čtyři aproximované body, dostaneme vztah pro výpočet Bezierovy kubiky:

$$P(t) = \sum_{i=0}^3 P_i B_i = P_0 B_0(t) + P_1 B_1(t) + P_2 B_2(t) + P_3 B_3(t) \quad (12)$$

Kde:

$P(t)$  – bod křivky, parametr  $t$  je z rozsahu  $\langle 0,1 \rangle$

$P_0$  – počáteční bod křivky

$P_1, P_2$  – body řídící vyklenutí křivky

$P_3$  – koncový bod křivky

$B_0(t), B_1(t), B_2(t), B_3(t)$  – Bernsteinovy polynomy stupně 3

Dle definice jsou Bernsteinovy polynomy dány vztahem

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (13)$$

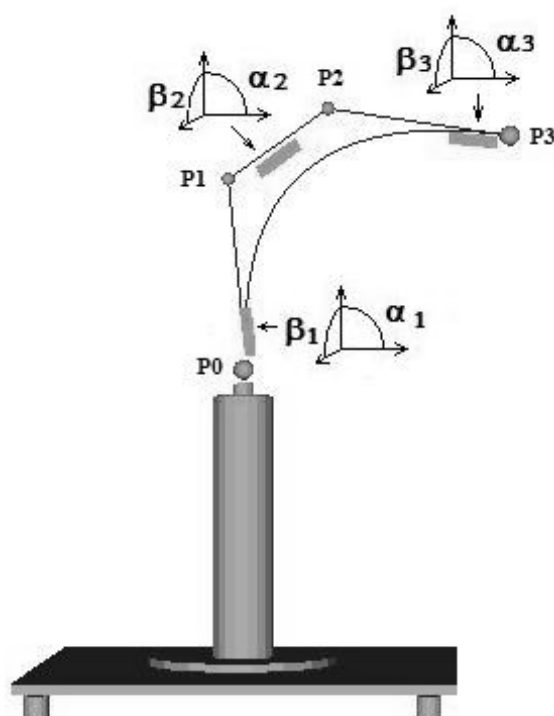
Rozepsáním pro polynomy stupně 0 až 3 dostáváme:

$$\mathbf{B}_0(t) = (1 - t)^3$$

$$\mathbf{B}_1(t) = 3t(1 - t)^2$$

$$\mathbf{B}_2(t) = 3t^2(1 - t)$$

$$\mathbf{B}_3(t) = t^3$$



Obr. 10: Modelování řídicího polygonu kubiky ve 3D užitím třech akcelerometrů

Dosazením Bernsteinových polynomů (13) dostáváme

$$\begin{aligned} \mathbf{P}(t) &= \sum_{k=0}^3 \mathbf{P}_i \mathbf{B}_i = \mathbf{P}_0 \mathbf{B}_0(t) + \mathbf{P}_1 \mathbf{B}_1(t) + \mathbf{P}_2 \mathbf{B}_2(t) + \mathbf{P}_3 \mathbf{B}_3(t) = (1-t)^3 \cdot \mathbf{P}_0 + 3t(1-t)^2 \cdot \mathbf{P}_1 + 3t^2(1-t) \cdot \mathbf{P}_2 + t^3 \cdot \mathbf{P}_3 \\ &= (-\mathbf{P}_0 + 3\mathbf{P}_1 - 3\mathbf{P}_2 + \mathbf{P}_3) \cdot t^3 + (3\mathbf{P}_0 - 6\mathbf{P}_1 + 3\mathbf{P}_2) \cdot t^2 + (-3\mathbf{P}_0 + 3\mathbf{P}_1) \cdot t + \mathbf{P}_0 \end{aligned}$$

Pro přehlednost přepíšeme do maticového tvaru:

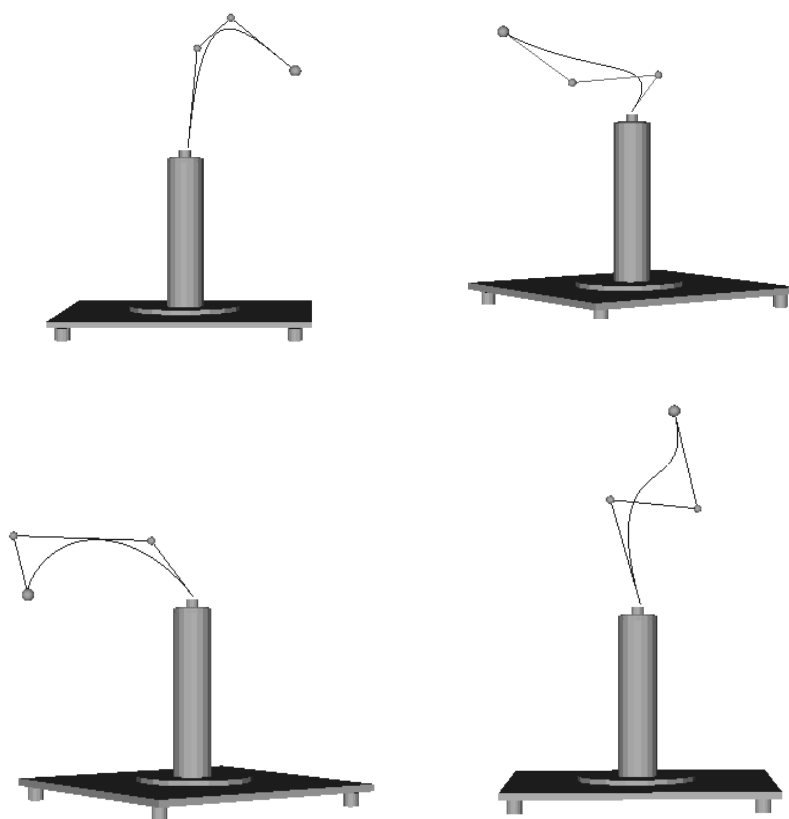
$$\mathbf{P}(t) = \underbrace{(t^3 \ t^2 \ t \ 1)}_{\text{vektor parametru } t} \underbrace{\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{\text{bázová matice}} \underbrace{\begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix}}_{\text{vektor bodů řídícího polygonu}} \quad (14)$$

Nakonec výraz rozepíšeme do směrů x, y a z:

$$P_x = (1-t)^3 \cdot P_{0x} + 3t(1-t)^2 \cdot P_{1x} + 3t^2(1-t) \cdot P_{2x} + t^3 \cdot P_{3x}$$

$$P_y = (1-t)^3 \cdot P_{0y} + 3t(1-t)^2 \cdot P_{1y} + 3t^2(1-t) \cdot P_{2y} + t^3 \cdot P_{3y}$$

$$P_z = (1-t)^3 \cdot P_{0z} + 3t(1-t)^2 \cdot P_{1z} + 3t^2(1-t) \cdot P_{2z} + t^3 \cdot P_{3z}$$



Obr. 11: Ukázky Bezierových kubik modelovaných řídícím polygonem se třemi senzory

## 1.6 Interpolace Catmull-Rom splajny

Dle definice maticový tvar Catmull-Rom splajnu vypadá následovně:

$$\mathbf{P}(t) = 0,5 \cdot \underbrace{(t^3 \ t^2 \ t \ 1)}_{\text{vektor parametru } t} \underbrace{\begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}}_{\text{bázová matice}} \underbrace{\begin{pmatrix} \mathbf{P}_{i-3} \\ \mathbf{P}_{i-2} \\ \mathbf{P}_{i-1} \\ \mathbf{P}_i \end{pmatrix}}_{\text{vektor bodů řídicího polygonu}} \quad (15)$$

Jelikož výsledný tvar dané křivky obecně neleží v konvexní obálce svých řídicích bodů, pro jeho generování byl použit řídicí polygon Bezierovy křivky. Díky dvojnásobnému výskytu prvního a posledního bodu ve vektoru řídicího polygonu první a poslední části křivky byla zajištěna interpolace křivky všemi řídicími body, včetně počátečního a koncového. Interpolační křivka je tak rozdělena na tři úseky, kde první je řízen body  $P_0, P_0, P_1, P_2$ , druhý  $P_1, P_2, P_3, P_4$  a třetí – body  $P_2, P_3, P_4, P_4$ . Roznásobením matice a vektorů (15) dostáváme:

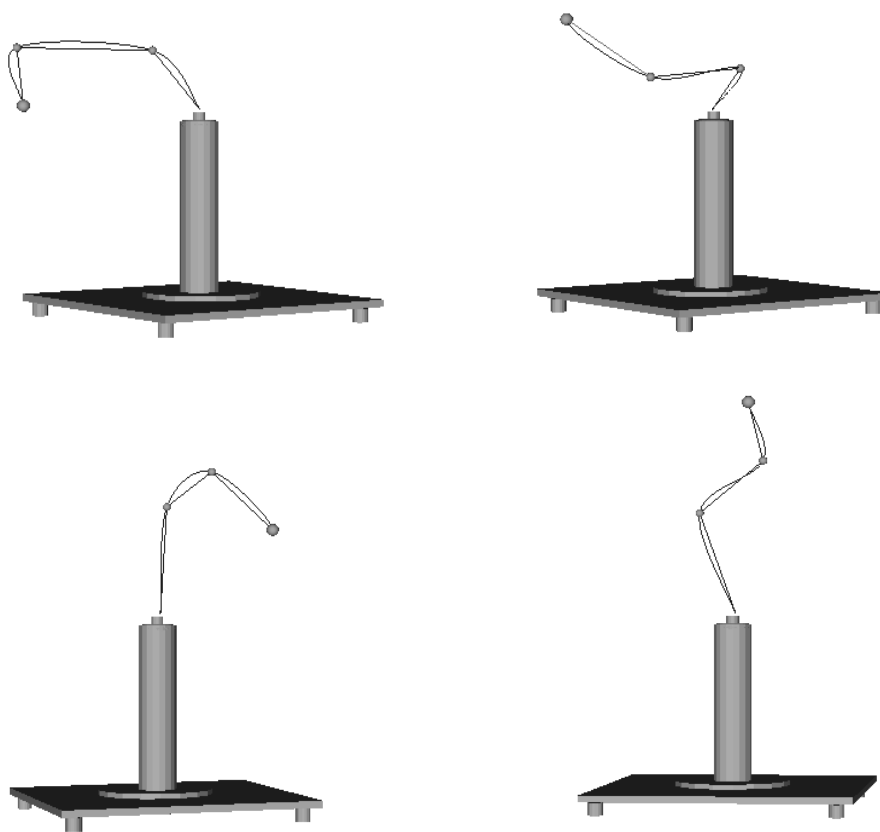
$$\mathbf{P}(t) = 0,5 \cdot (-P_1 + 3P_2 - 3P_3 + P_4)t^3 + (2P_1 - 5P_2 + 4P_3 - P_4)t^2 + (-P_1 + P_3)t + 2P_2$$

Rozepsáním do směrových složek získáme vztah pro výpočet křivky v 3D:

$$P_x = 0,5 \cdot (-P_{1x} + 3P_{2x} - 3P_{3x} + P_{4x})t^3 + (2P_{1x} - 5P_{2x} + 4P_{3x} - P_{4x})t^2 + (-P_{1x} + P_{3x})t + 2P_{2x}$$

$$P_y = 0,5 \cdot (-P_{1y} + 3P_{2y} - 3P_{3y} + P_{4y})t^3 + (2P_{1y} - 5P_{2y} + 4P_{3y} - P_{4y})t^2 + (-P_{1y} + P_{3y})t + 2P_{2y}$$

$$P_z = 0,5 \cdot (-P_{1z} + 3P_{2z} - 3P_{3z} + P_{4z})t^3 + (2P_{1z} - 5P_{2z} + 4P_{3z} - P_{4z})t^2 + (-P_{1z} + P_{3z})t + 2P_{2z}$$



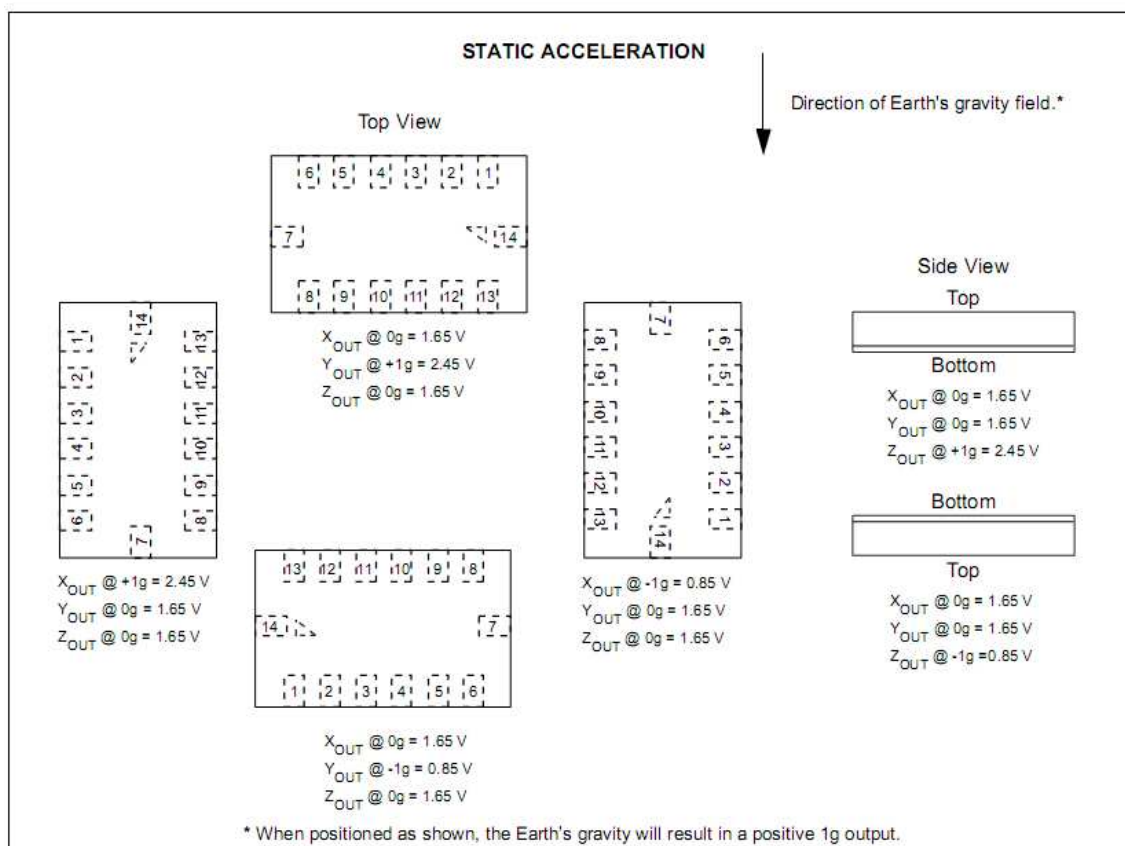
*Obr. 12: Ukázky Catmull-Rom splajnů modelovaných řídicím polygonem se třemi senzory*

## 1.7 Porovnání metod modelování křivek

Experimenty s výběrem vhodné interpolační nebo aproximační metody ukázali neuniverzalitu zvoleného postupu modelování křivek, vhodná křivka se musela volit individuálně v závislosti na vlastnostech použitého ohebného spojení mezi senzory. Obecně lze však říci, že pro tenčí a ohebnější spojení je vhodnější doplnění úseků mezi body pomocí interpolace s větším počtem senzorů. Méně ohebné spojení zkonstruovaného přípravku přesněji napodobovaly aproximační metody.

## 1.8 Omezení vybraných metod měření

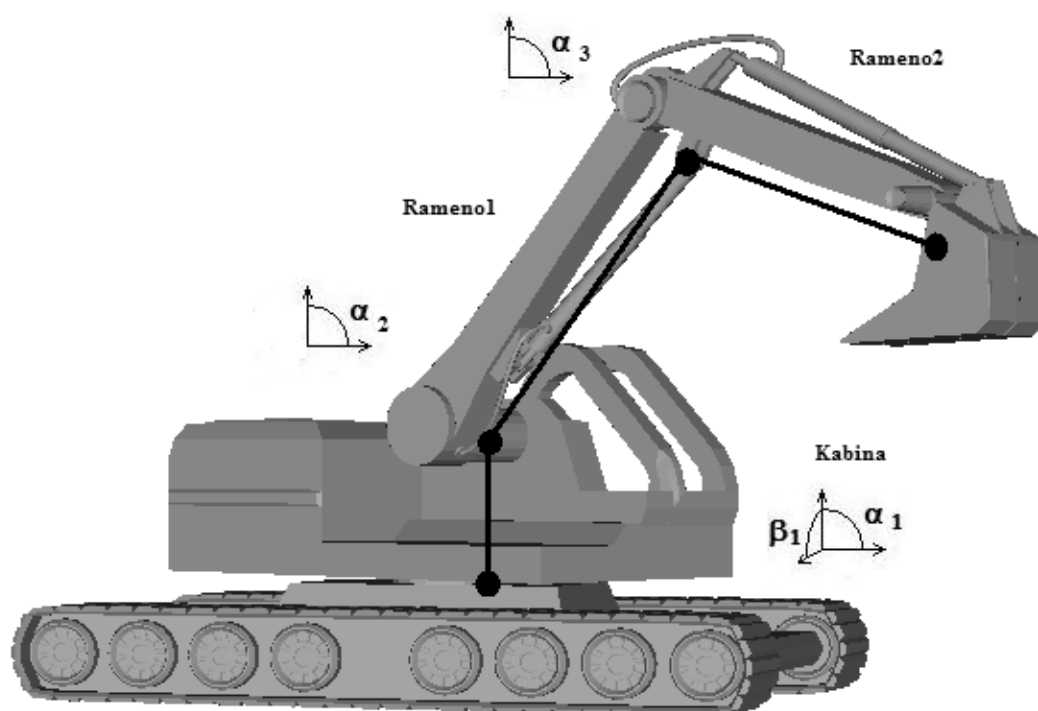
V minulé kapitole jsme se již krátce zmínili o omezeních měření při kloubovém a ohebném spojování senzorů. Dalším problémem je podmínka umístění měřené křivky ve svislé poloze. Dané omezení je způsobeno samotným principem měření statického zrychlení pomocí akcelerometrů, nedovolujícím měření natočení senzoru kolem os paralelních k vektoru zemské gravitace (viz obrázek č. 13).



Obr. 13: Způsob měření statického zrychlení akcelerometru (zdroj: Freescale, datasheet akcelerometru MMA7361L)

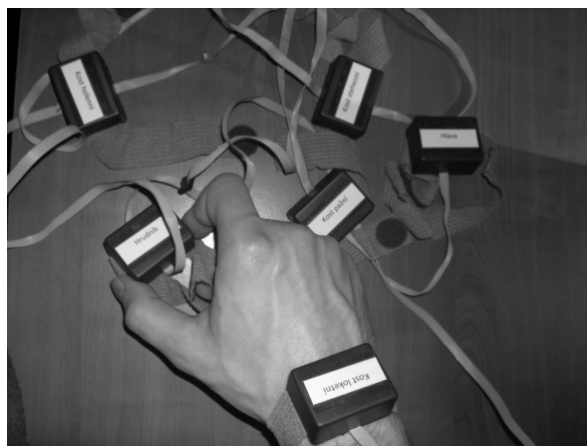
## 1.8 Měření křivek kloubově spojenými senzory

Hlavním problémem při realizaci tohoto způsobu měření byla absence vhodných kloubových mechanických součástí. V průběhu hledání řešení bylo pro účely modelování kloubového spojení nejprve používáno zařízení se třemi akcelerometry, znázorněné na obrázku č. 8. Modelování bylo předvedeno na modelu bagru. Otáčení kabiny kolem osy kolmé k podvozku a tudíž paralelní vektoru působení zemské gravitace bylo realizováno metodou měření odchylky od svislice za předpokladu, že senzor je pevně připevněn na otáčejícím se nakloněném rameni. Otáčení ramene bagru je modelováno otáčením senzorů číslo dva a tři.



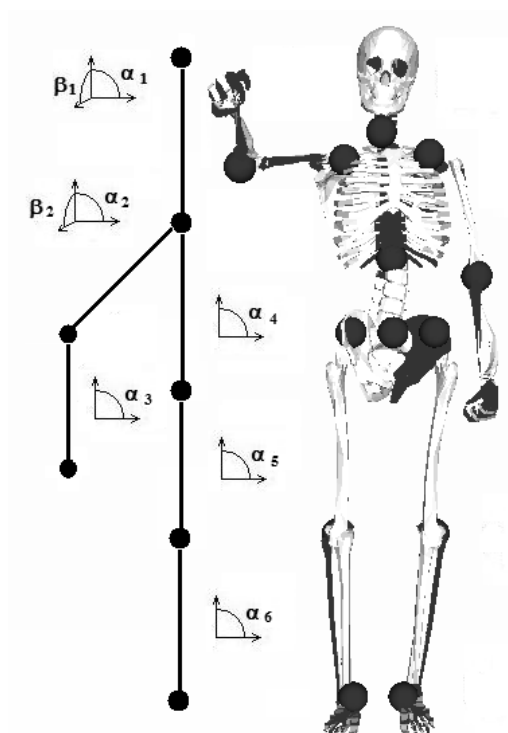
Obr. 14: Ukázka měření kloubově spojenými senzory.

Pro účely modelování křivek daných kloubově spojenými senzory vzniklo zařízení zobrazené na obrázku č. 15. Hardwarově se jedná o stejné moduly, jaké byly použity v prvním zařízení, spojené do sítě.



*Obr. 15: Fotografie přípravku pro měření křivek daných kloubově spojenými senzory*

Senzory lze například rozmístit po těle a měřit jeho polohu a pohyby, podrobněji viz kapitola 3.



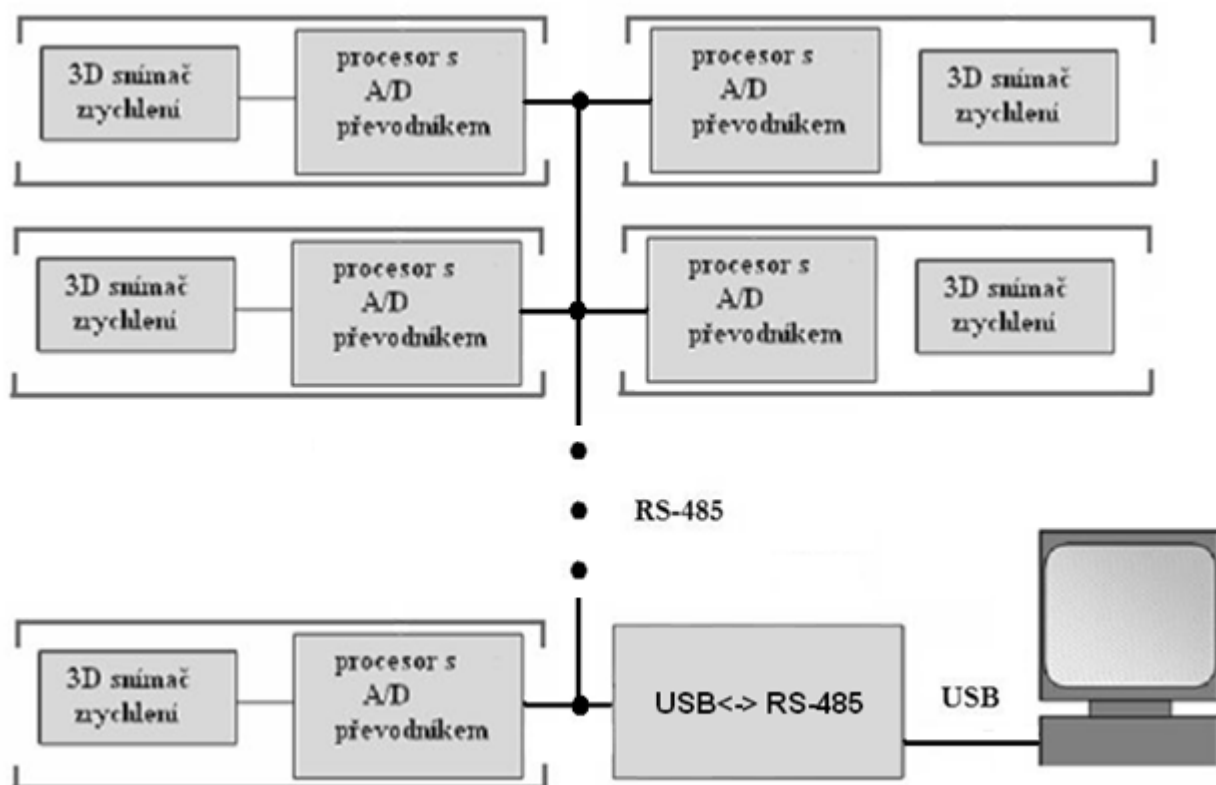
*Obr. 16: Schéma přípravku pro měření křivek daných kloubově spojenými senzory a model lidské kostry*



## 2 Hardware

### 2.1 Ideové schéma

Na obrázku níže je znázorněno ideové schéma zařízení. S počítačem systém komunikuje přes USB. Ke komunikaci v systému slouží sběrnice RS-485. Data přijímaná z počítače jsou řídicím modulem, který je v daném okamžiku v roli master, rozesílána po síti. Modul, jehož adresa byla řídicím modulem odeslána, po jejím přijetí přebírá řízení sítě na sebe, ze stavu slave přechází do stavu master a rozesílá po síti naměřená data.

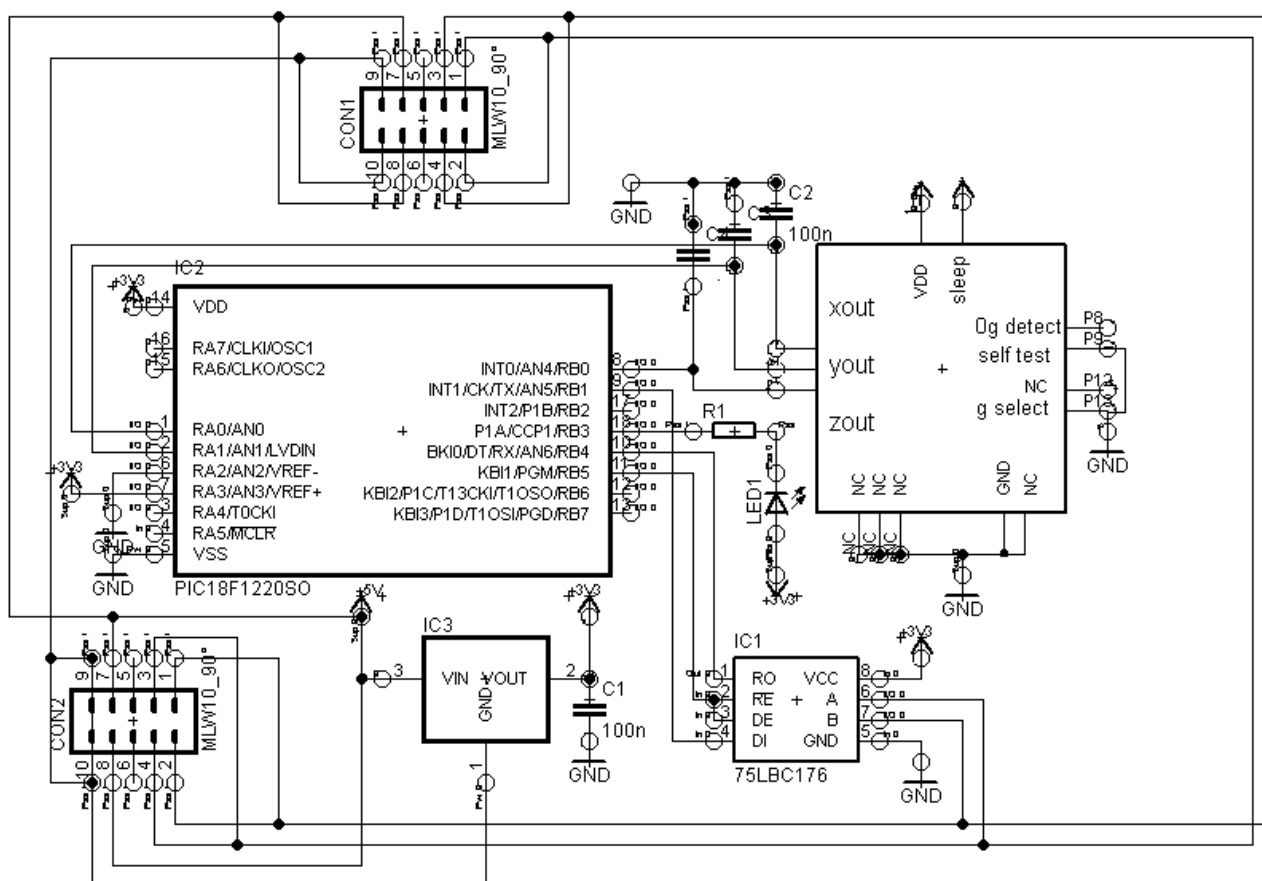


*Obr. 17: Ideové schéma zařízení*

Celý systém je napájen přes USB. Transceivery 75LBC176, použité pro sběrnici RS-485, dovolují připojit k síti až 256 zařízení. Proto hlavním kritériem při návrhu opakujících se částí systému byla jejich cena a nízký proudový odběr.

## 2.2 Hardware modulů s akcelerometry

Dole lze vidět schéma navržených modulů s akcelerometry:



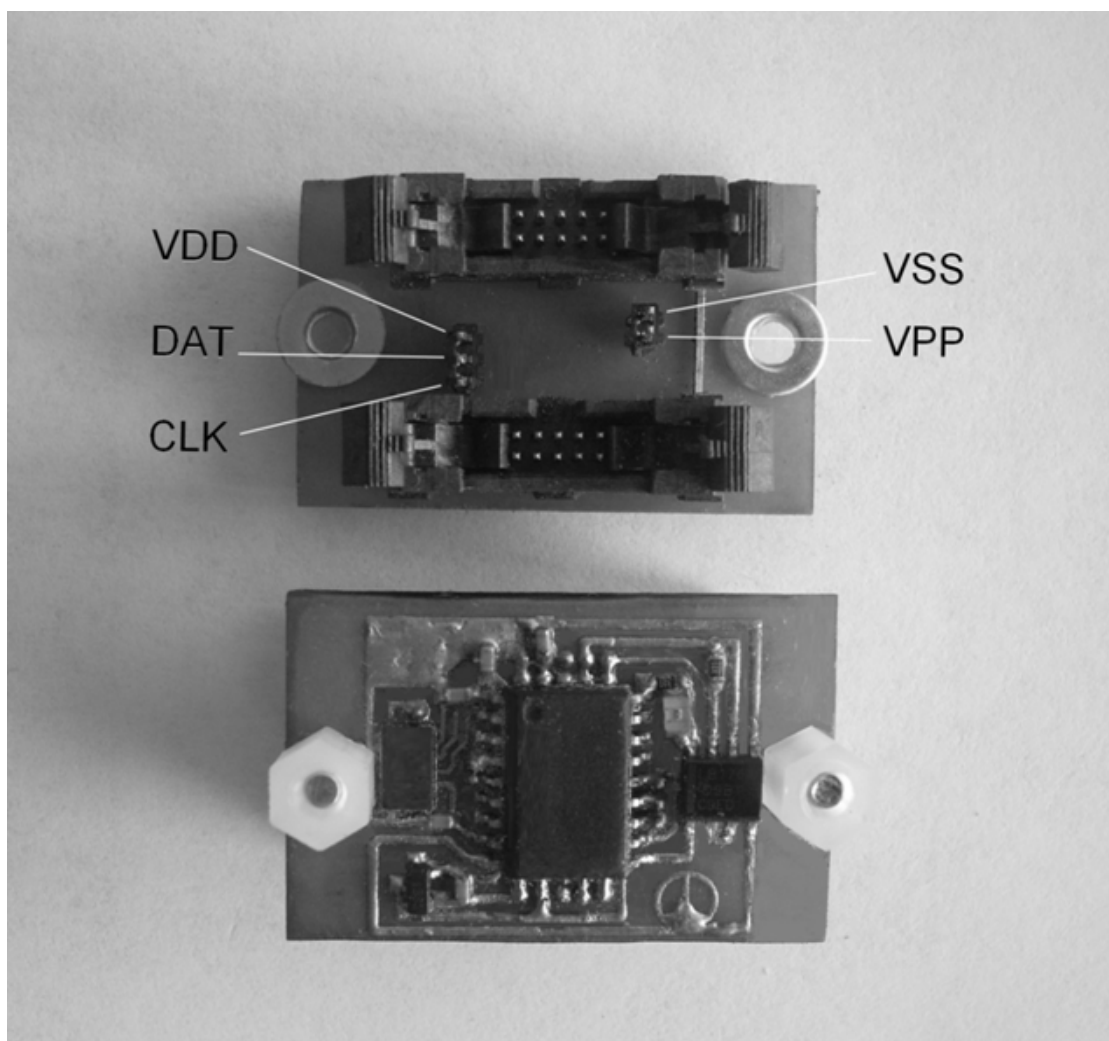
Obr. 17: Schéma modulů s akcelerometry

Pro měření napětí na výstupech akcelerometru byl použit A/D převodník mikrokontroleru PIC18LF1220. Rozlišení analogově digitálního převodníku je dáno vztahem. V našem případě bylo referenční maximum  $U_{max}$  nastaveno na 3,3 voltů a minimum  $U_{min}$  na nulu.

$$\Delta U = \frac{U_{max} - U_{min}}{2^b - 1}$$

Pro případ osmibitového převodníku je rozlišení převodníku 12,94mV. Dle datasheetu akcelerometru MMA7361L (obr. 7) by měl být rozsah výstupních hodnot senzoru v rozmezí 0,85V až 2,45V. Ve skutečnosti se offse-

ty os často lišily až o desítky milivoltů a bylo třeba akcelerometry individuálně kalibrovat. Stejně tak se musely kalibrovat i interní RC oscilátory na přesných 8MHz, nutných pro bezchybnou komunikaci v síti. Napětí 5V z portu USB je upravováno nízkoúbytkovým stabilizátorem MCP1700-3.3 na napětí 3,3V. Pro indikaci činnosti procesoru slouží LED dioda. Pro programování procesoru přímo v aplikaci je vyveden ICSP port.



*Obr. 18: Programovací piny modulu*

Deska byla navržena v programu Eagle, je jednostranná a má rozměry 25 x 20 mm.

## 2.3 Firmware modulů s akcelerometry

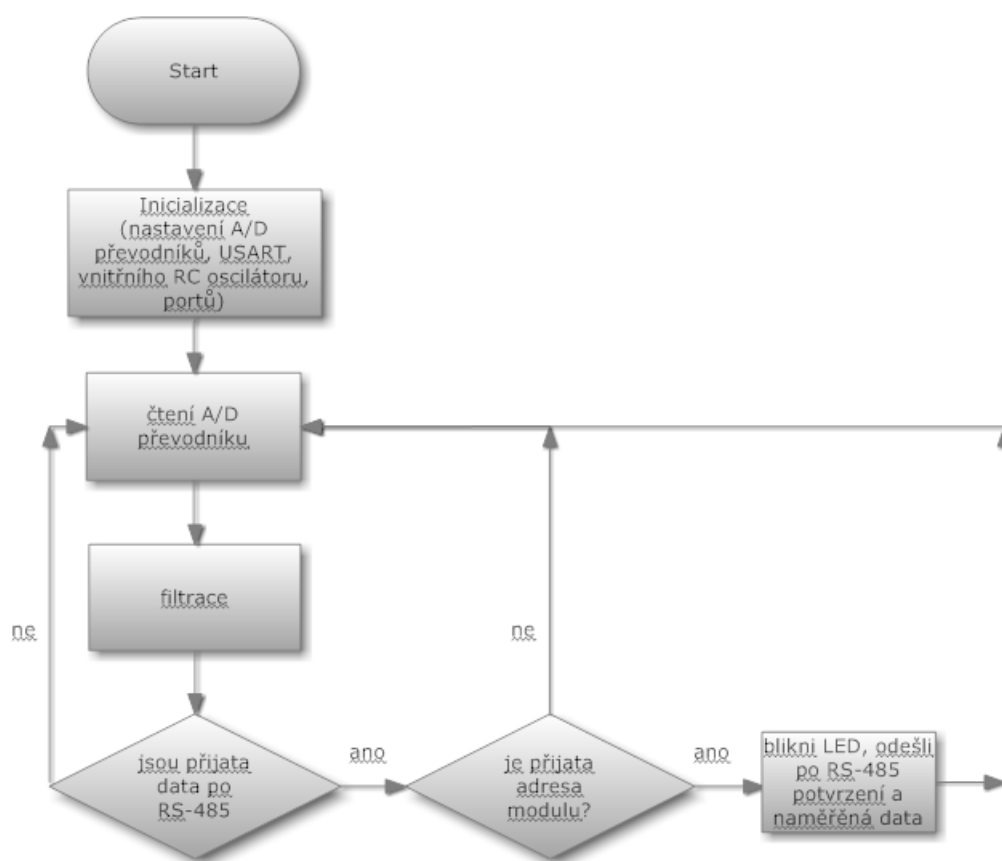
Firmware modulů s akcelerometry a USB/RS-485 převodníku byl psán ve studentské verzi kompilátoru C18. Program pro moduly je rozdělen do třech částí: inicializace, čekání na přijetí adresy, čtení A/D převodníku, filtrace a vysílání po RS-485. Při inicializaci je kromě počátečního nastavení portů, USART modulu a A/D převodníku provedeno nastavení vnitřního RC oscilátoru na 8MHz, případně korekce této hodnoty na požadovanou pomocí registru OSCTUNE. Neobvyklým řešením bylo v režimu čekání nenechávat procesor ve stavu nízké spotřeby, ze kterého by byl probouzen až přijetím své adresy, ale nechat ho celou dobu pracovat na plný výkon: procesor stále čte A/D převodník a filtruje získávané hodnoty. Tím je dosaženo dokonalé redukce šumu, který bez filtrace u akcelerometrů MMA7361L standardně dosahuje hodnot až  $\pm 10^\circ$  a zároveň zachována vysoká rychlost reakce na prudké změny natočení senzoru. Filtrovaný signál je počítán v plovoucí desetinné čárce, je filtrován IIR filtrem, daným vztahem:

$$y_n = k(x_n) + (1 - k)y_{n-1}$$

Po přijetí příslušné adresy modulu je odeslána krátká odpověď ve tvaru:

Adresa modulu	X data	Y data	Z data
---------------	--------	--------	--------

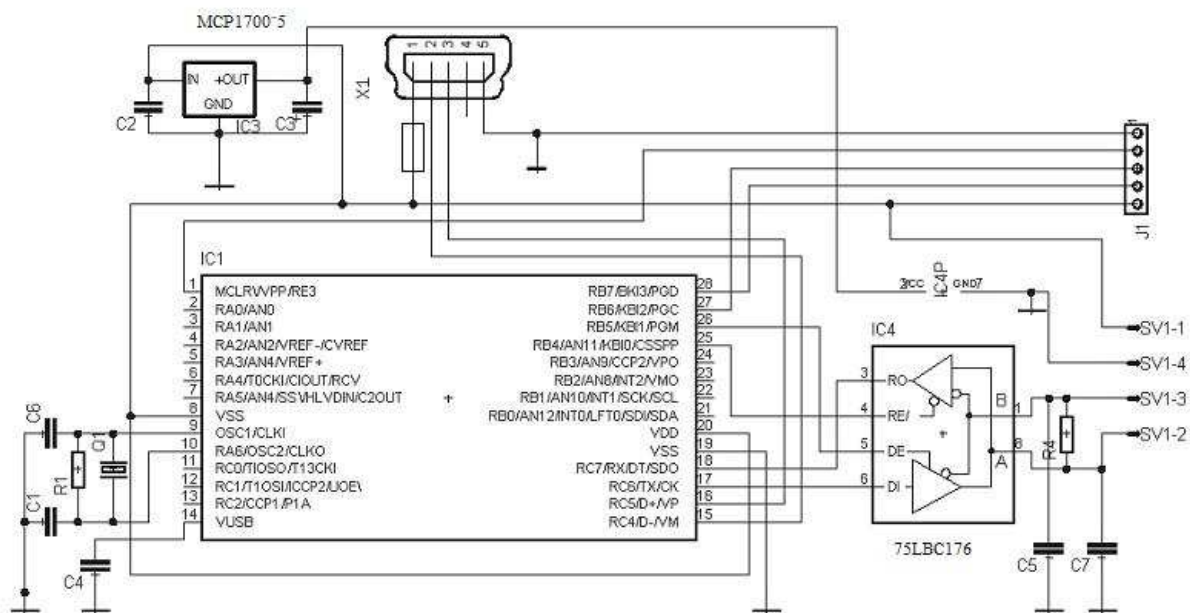
Hlavička zprávy informuje, kterému senzoru data patří. Zpáteční adresa modulu je pro kontrolu upravena přičtením čísla 10. Odeslání je indikováno bliknutím LED diody.



Obr. 19: Vývojový diagram modulů

## 2.4 Hardware převodníku USB/RS-485

Jak je ze schématu na další stránce vidět, jádrem zařízení je procesor PIC18LF2550. Procesor je taktován frekvencí 48MHz, získávané ze 12MHz krystalu pomocí vnitřní násobičky hodin. Pro stabilizaci napětí USB sběrnice byl použit integrovaný v procesoru regulátor napětí na 3,3 V. Regulátor MCP1700-5 slouží pro stabilizaci napětí 5V, distribuovaného po sběrnici. Proudová ochrana je realizována vrátnou pojistkou na 140mA.



Obr. 20: Schéma převodníku USB/RS-485

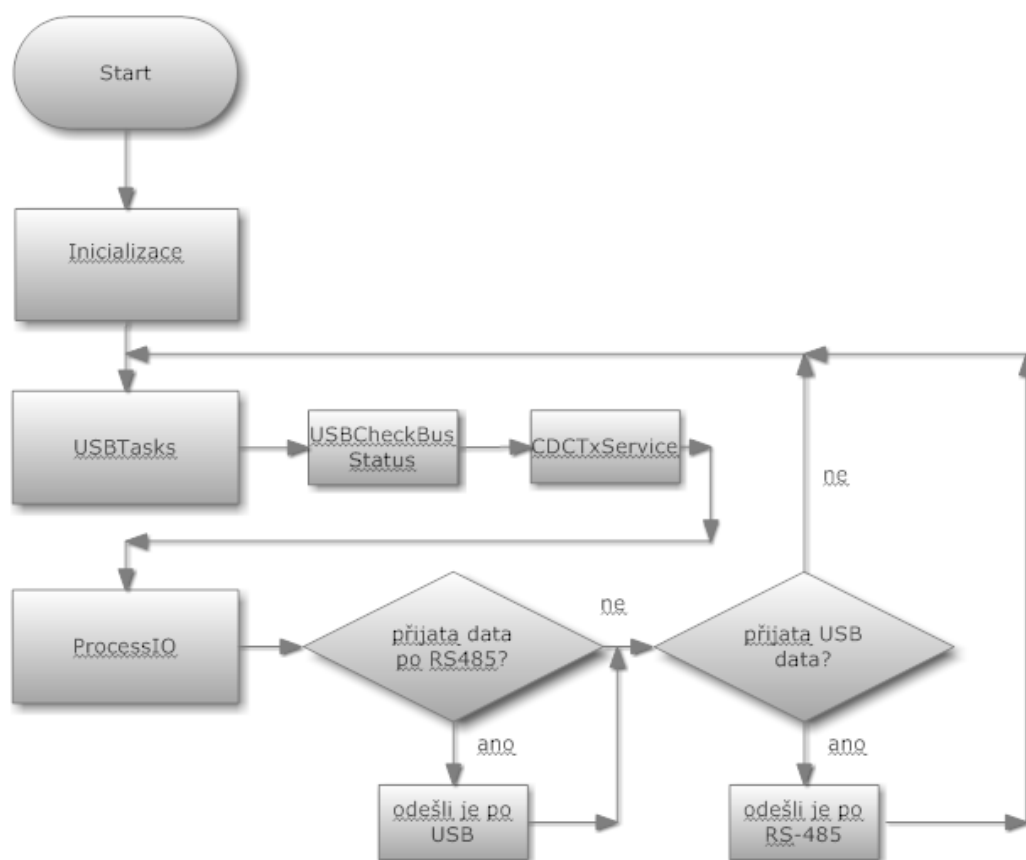
Zvláštní pozornost při návrhu desky byla, dle doporučení výrobce, věnována stínění hodinových kmitů krystalu. Modul EUSART je nakonfigurován pro práci rychlostí 9600 baudů v asynchronním plně duplexním režimu.



Obr. 21: Fotografie převodníku USB/RS-485

## 2.5 Firmware převodníku USB/RS-485

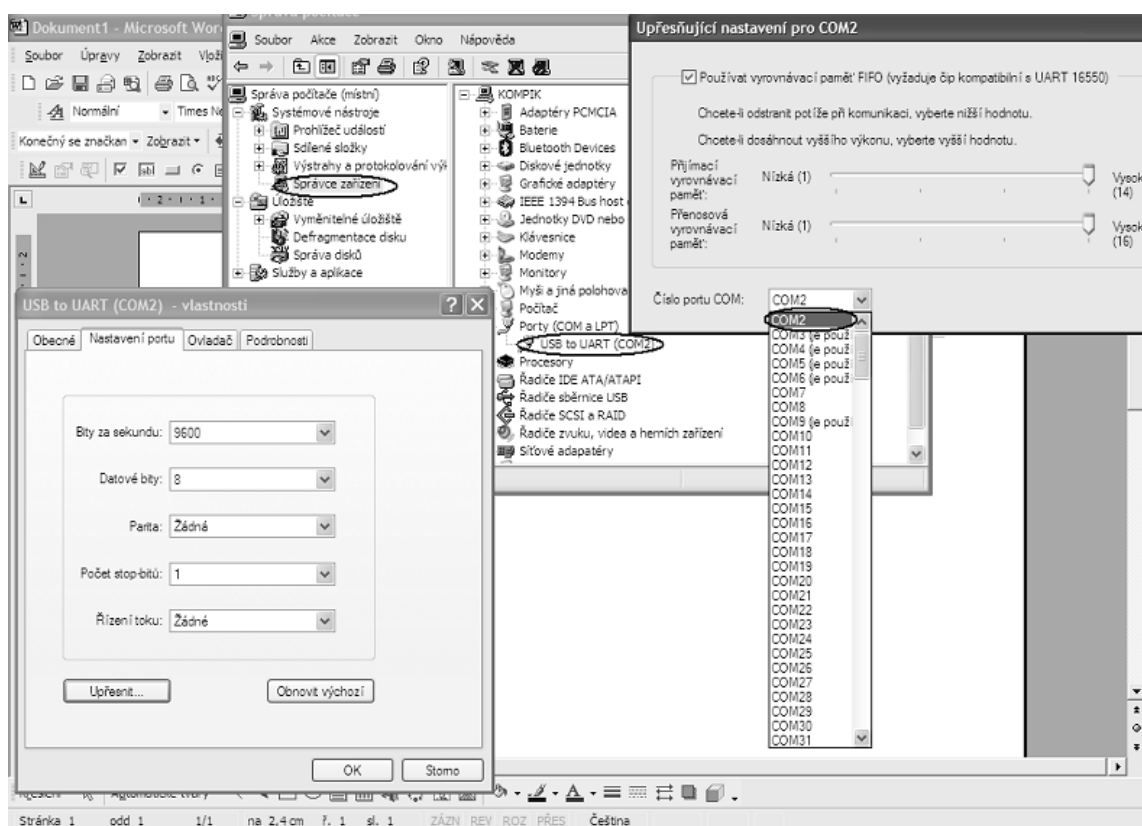
Na obrázku je znázorněn vývojový diagram programu. Po inicializaci mikrokontroler průběžně vykonává obsluhu USB komunikace, kontroluje příznaky naplnění přijímacího modulu EUSART a příznaky přijetí dat po USB. V nekonečném cyklu jsou přijímaná po USB data přímo přeposílána po RS-485 a opačně, přijímaná po RS-485 data jsou sestavována do paketů a přeposílána po USB do PC.



Obr. 22: Vývojový diagram převodníku USB/RS-485

Při výběru typu USB komunikace jsme volili mezi HID (Human Interface Device), zařízením s ovládačem (Custom Driver Device) a emulaci sériového portu (Communication Device Class). Byl zvolen třetí typ komunikace. Výhodou komunikace v režimu emulace sériového portu oproti zařízením s ovládačem a HID je přehledná ovladatelnost zařízení počítačem.

Zařízení emuluje sériový port, proto ke komunikaci počítačové aplikace s USB zařízením lze použít standardní knihovny pro RS-232. Daný způsob komunikace lze, na rozdíl od ostatních metod USB komunikace, považovat za multiplatformní, s podmínkou, že v operačních systémech Windows je název virtuálního portu přiřazován po připojení zařízení k USB (lze ho měnit ve vlastnostech portu) a tento údaj si daný USB port pak „pamatuje“. V Linuxu je připojený virtuální sériový port rozlišován od reálného, je označován dev/tty/ACM, kdežto reálné porty jsou označovány dev/tty/S+číslo portu. Pro komunikaci s CDC zařízením v Linuxu navíc nebylo potřeba instalovat ovládač. Po prvním připojení zařízení ve Windows počítač nahlásí nové neznámé zařízení a měla by proběhnout jeho instalace (ovládač má název MchpUSB.inf, lze ho najít na přiloženém CD ve složce „ovladace“).



Obr. 23: Ukázka nastavení čísla virtuálního sériového portu ve Windows.



Číslo portu by mělo být v rozmezí 1 až 4, lze ho nastavit v „Tento počítač“ -> „Spravovat“ -> „Správce zařízení“ -> ve vlastnostech nainstalovaného portu -> „Nastavení portu“-> „Upřesnit“.

## 3 Software

### 3.1 Metody zobrazování 3D modelů

Jak už jsme řekli, při modelování křivek je ke správné interpretaci naměřených údajů potřeba vycházet z rozměrů těles, na kterých budou senzory umístěny a které bude křivku reprezentovat. Nástrojem pro tvorbu modelů byl zvolen standard OpenGL (*Open Graphics Library*) specifikující multiplatformní rozhraní pro tvorbu aplikací počítačové grafiky a často se používá se při tvorbě vědeckotechnické vizualizace, CAD programů, aplikací virtuální reality, apod. Tvorba modelů v OpenGL bez podpůrných prostředků je dosti pracná a nepřehledná. Proto bylo třeba vymyslet dostatečně jednoduchý způsob jak je tvořit. Pro účel tvorby modelů ve 3D existuje celá řada specializovaných nástrojů, asi nejznámějším z nich je program Autodesk 3ds Max. Tato aplikace podporuje export modelů do celé řady souborových formátů – kromě nativních souborů s příponou MAX jsou to i nejrůznější CAD formáty apod. Zaměřili jsme se na export do textových souborů. Prvním z nich byl soubor typu ASE (ASCII Scene Export). Při pokusech zpracovávání a identifikace obsahu souborů tohoto typu v Delphi jsem narazil na problém proměnlivé mezery mezi sloupci hodnot. Problém lze pozorovat vlevo dole na příkladu exportovaného modelu krychle ve formátu ASE, ze kterého se nepodařilo jednoznačně vyvodit podmínku vzniku oddělovací mezery mezi sloupci.



## Struktura GMS souborů:

---

```
New object          // Tento výraz označuje začátek nového objektu.

TriMesh()           // Zde se nachází typ objektu- v daném případě síť.

numverts numfaces    // Za tímto výrazem se nachází počet vrcholů a hran
objektu.

Mesh vertices:

    ...              // Blok souřadnic vrcholů objektu ve tvaru X Y Z.
end vertices

Mesh faces:
    ...              // Zde se nachází blok hran objektu ve tvaru indexů
                      // 1, 2 a 3, každý index je indexem pro pole vrcholů,
                      // ukazující na vrchol příslušné hrany.
end faces

Faset normals:
    ...              // Blok normál ve tvaru souřadnic X Y Z, jejich po-
                      // čet je roven počtu ploch, tvořených hranami předcho-
                      // zího bloku.
end faset normals

end mesh             // Konec popisu objektu
end of file           // Konec souboru
```

---

Bloků, obsahujících danou strukturu může být v jednom souboru i více, pokud konvertovaný model byl složen z více než jedné části. Proto před exportem modelu ze 3D studio Max je výhodné model nejprve spojit příkazem Collapse selected, zaškrtnout v „Output type“ radiobutton „Mesh“ a v nabídce „Collapse to“ zvolit „Single object“. Export modelů ze 3D studio Max se provádí v nabídce MAXscript. Tam v „Utilities“ zvolíme položku Mega, zaškrtneme volbu „selected only“ (před tím nezapomeneme označit model, který chceme exportovat), ve „Frames“ zadáme „From: 0“, „To: 1“ a soubor s příponou \*.GMS uložíme.

Načtení obsahu souborů \*.GMS v Delphi pak může vypadat například následovně (jedná se o postup, navržený pro účely dané práce, avšak lze ho použít i kdekoli jinde, kde bude potřeba pracovat se složitými modely, proto si ho dovolíme krátce popsat).

```
AssignFile(f,'stojank.gms'); // přiřad' název souboru
Reset(f);                    // načti soubor
Repeat                       //opakuj čtení
  ReadLn(f, S);              //dokud nenarazíš na text 'numverts numfaces' nebo na konec souboru
until (S = 'numverts numfaces') or eof(f);

ReadLn(f,VertexCount,FacesCount); // načti celkový počet vrcholů a ploch

GetMem(Vertices,VertexCount*SizeOf(TGLVertex)); // připrav místo pro vrcholy
GetMem(Faces,FacesCount*SizeOf(TGLFace));        // připrav místo pro hrany
GetMem(FasetNormals,FacesCount*SizeOf(TGLVector)); // připrav místo pro normály

ReadLn(f, S); // přeskoč řádek "mesh vertices"

  for i := 0 to VertexCount - 1 do // v cyklu načítej vrcholy
  begin
    ReadLn(f, Vrchol.x, Vrchol.y, Vrchol.z);
    Vrcholy[i] := Vrchol; // ukládej je do proměnné Vrchol
  end;
ReadLn(f, S);

ReadLn(f, S); // přeskoč řádky "end vertices" a "Mesh faces:"

  for i := 0 to FacesCount - 1 do // v cyklu načítej hrany
  begin
    ReadLn(f,Hrana[0], Hrana [1], Hrana [2]);

    Hrana[0] := Hrana[0] - 1;
    Hrana[1] := Hrana[1] - 1;
    Hrana[2] := Hrana[2] - 1;

    Plocha[i] := Hrana; // a ukládej je, zmenšené o 1, do proměnné Plocha
  end;
```

```

ReadLn(f, S);                                // přeskoč řádek „end faces“
ReadLn(f, S);                                // přeskoč řádek „Faset normals:“
  for i := 0 to FacesCount - 1 do begin      // v cyklu načítej normály
    ReadLn(f,Normala.x,Normala.y,Normala.z);
    Normaly[i] := Normala;                   // a ukládej je do proměnné Normal
  end;
CloseFile(f);                                // zavři soubor

```

Načítání souborů lze vykonávat i cyklicky, díky tomu lze pracovat s modely přímo ve 3D Max a současně, v reálném čase, pozorovat jejich změny, zobrazované ve stále běžící aplikaci v Delphi, bez nutnosti kompilovat. Užitím funkcí OpenGL pak vykreslování daným způsobem načteného modelu může vypadat velmi jednoduše:

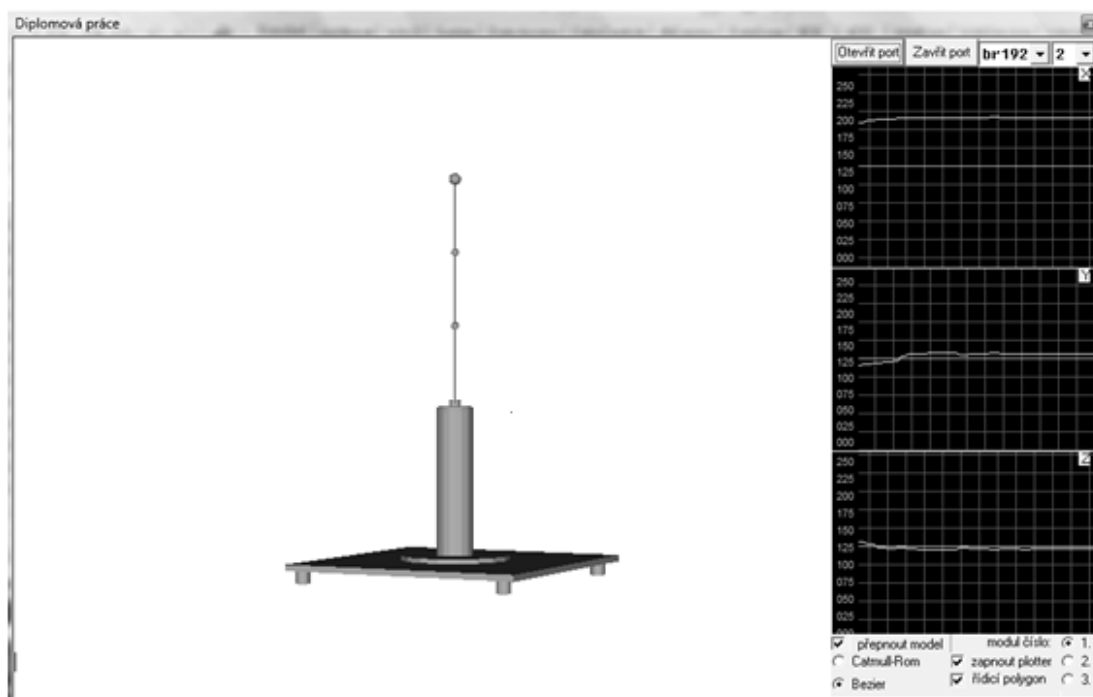
```

for i := 0 to FacesCount - 1 do begin        // až do celkového počtu ploch
  glBegin(GL_TRIANGLES);                     // kresli trojúhelníky:
    begin
      glNormal3fv(@Normaly[i]);              // definuj příslušnou normálu
      glVertex3fv(@Vrcholy[Hrana[0]]);      // definuj 3 vrcholy příslušné plochy
      glVertex3fv(@Vrcholy[Hrana [1]]);
      glVertex3fv(@Vrcholy[Hrana [2]]);
    end;
  end;
glend();

```

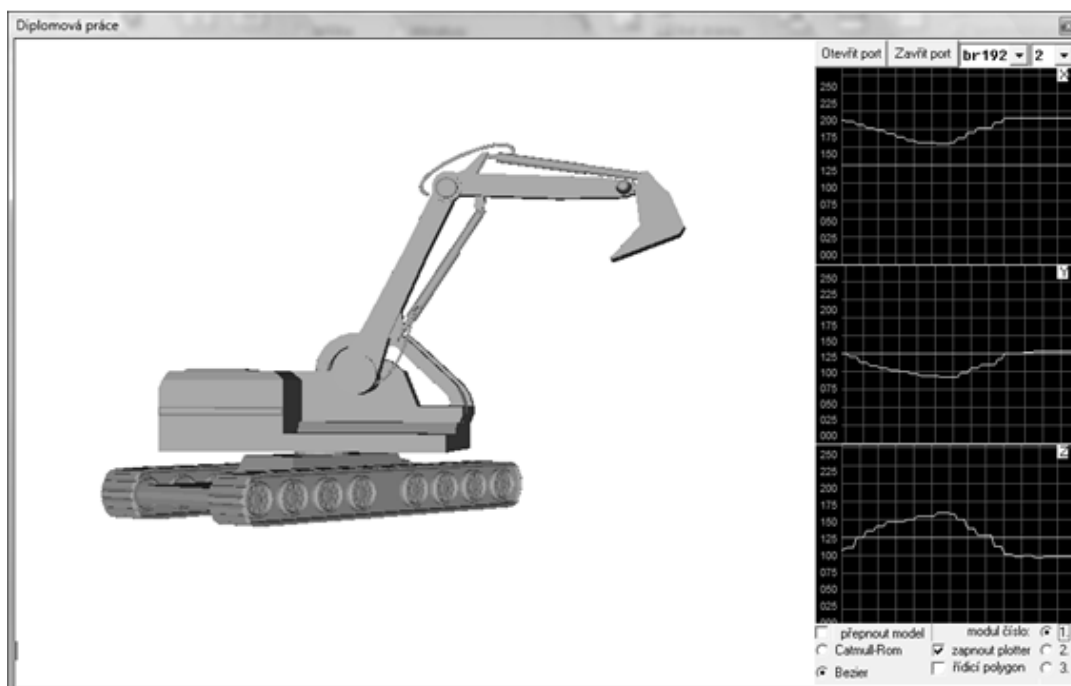
## 3.2 Popis počítačových aplikací

Pro dva zkonstruované přípravky byly napsány dvě počítačové aplikace. Aplikace, jejíž hlavní okno lze vidět na obrázku č. 25, slouží pro ovládání stojanu s akcelerometry, popsaného v kapitole 1.4.



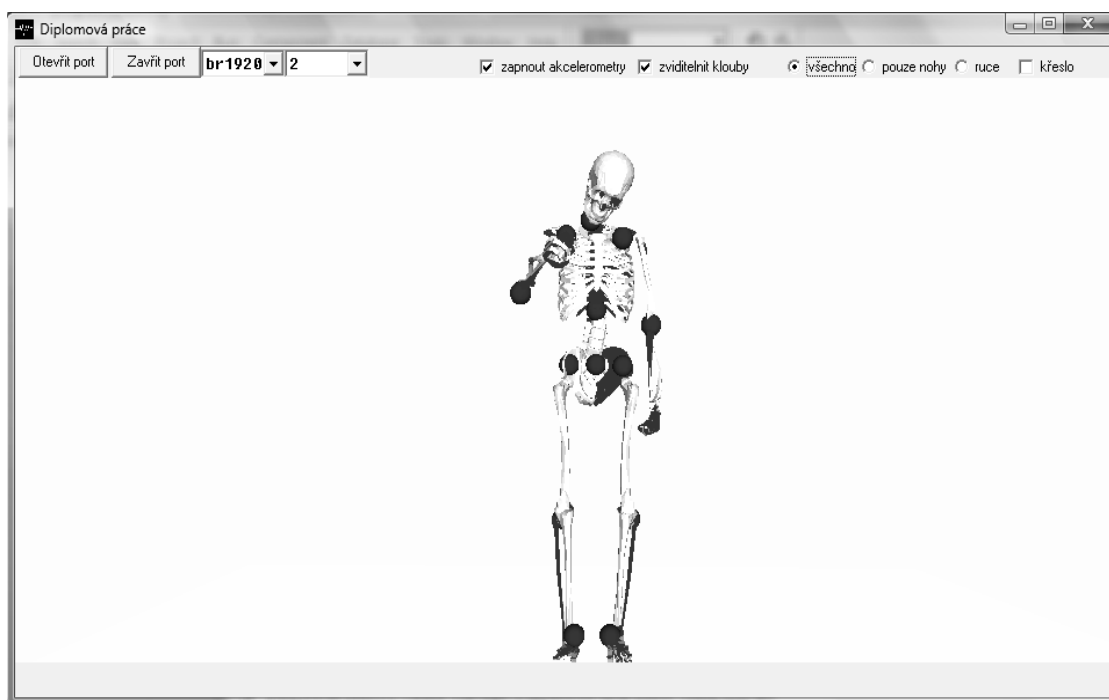
Obr. 25: Hlavní okno aplikace pro přípravek s ohebně spojenými akcelerometry.

V pravé části okna jsou tři prvky pro vykreslování dat, přijímaných ze senzorů. Jejich činnost je řízena checkboxem „zapnout plotter“. Přepínáním radiobuttonů Catmull-Rom a Bezier lze volit mezi interpolací a aproximací křivky. K zobrazení řídicího polygonu křivky slouží checkbox „řídící polygon“. Přepínání mezi modely stojanu a bagru se řídí checkboxem „přepnout model“.



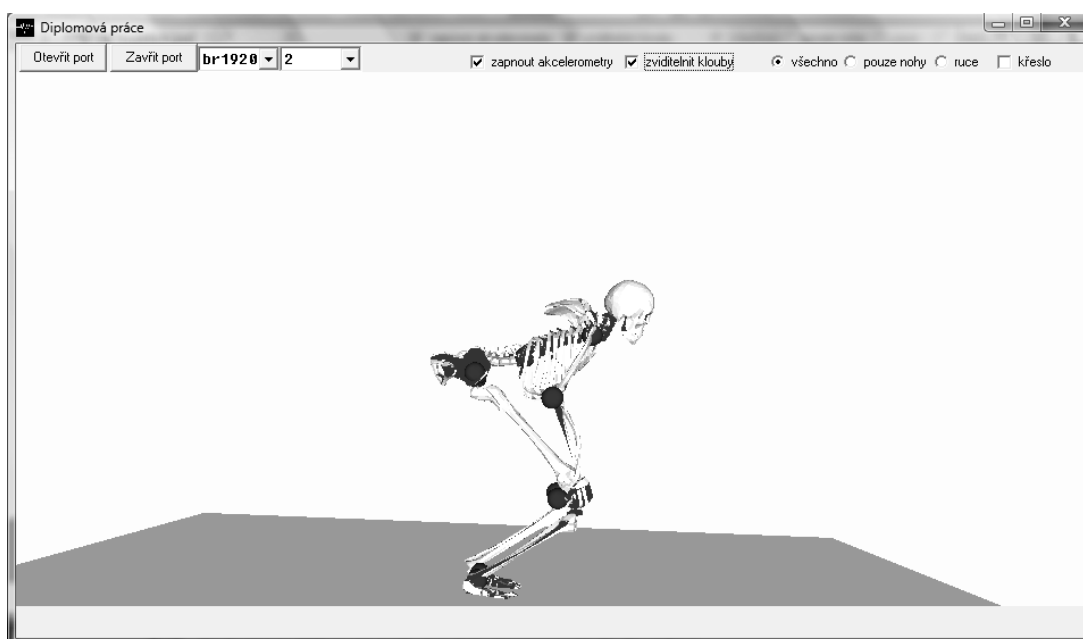
Obr. 26: Příklad užití programu pro modelování kloubově spojených objektů pomocí stojanu s akcelerometry.

Otáčení modelů v prostoru se provádí pravým tlačítkem myši, přibližování a oddalování se řídí točením kolečka myši. Pro komunikaci s USB převodníkem lze použít jakoukoliv komponentu pro RS-232 komunikaci. Pro sériovou komunikaci v Delphi byla použita komponenta VaComm. Po aktivaci tlačítka „Otevřít port“ program otevírá port, jehož číslo lze volit v comboboxu nahoře vpravo, a začíná rozesílání požadavků o vyslání naměřených dat. Požadované adresy modulů se generují cyklicky v rozsahu od nejnižší po nejvyšší. Combobox pro výběr rychlosti komunikace (baudrate), která nehraje v případě užití virtuálního sériového portu roli, je ponechán pouze pro kompatibilitu při případné komunikaci s RS-232 zařízeními. Další aplikací, vytvořenou pro modelování křivek daných kloubově spojenými akcelerometry lze vidět na obrázku č. 27, k její činnosti bylo použito zařízení, sestavené ze síťově spojených měřících modulů v krabičkách, umožňujících jejich připevnění k lidskému tělu, zmíněné v kapitole 1.8.



Obr. 27: Příklad užití programu pro modelování lidské kostry.

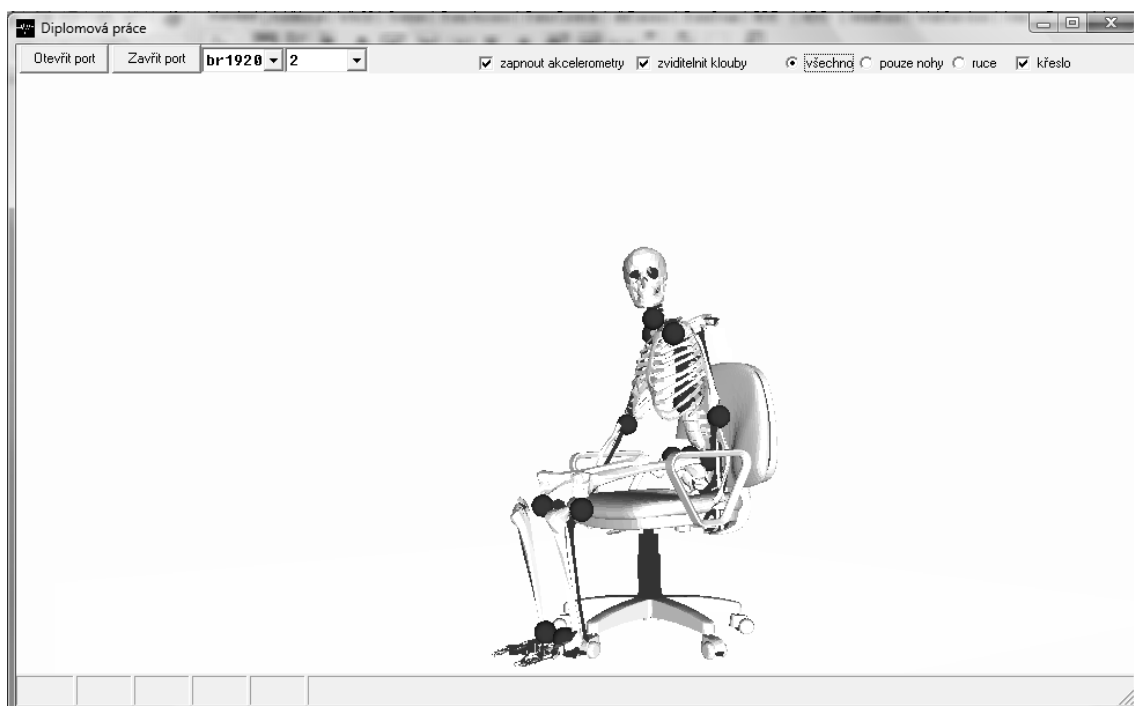
Při tvorbě pohyblivé lidské kostry byly použity open source 3D modely kostí, vzniklé v rámci projektu Visible Human, se kterým se lze blíže seznámit například na adrese <http://www.nlm.nih.gov/research/visible/>. Modely ve formátu MAX byly nejprve exportovány do souborů GMS a již popsaným způsobem použity v Delphi.



Obr. 28: Checkbox “zviditelnit klouby” zapíná vykreslování kloubů.



Checkbox „zviditelnit klouby“ zapíná zobrazování červených bodů v oblasti kloubů. Checkbox „křeslo“ zobrazí model křesla:



Obr. 29: Checkbox “křeslo” zapíná vykreslování křesla.

V literatuře [3] jsem narazil na nejrozumnější metody měření otáčení kolem os kolmých k vektoru zemské gravitace, například měřením intenzity ultrazvuku vysílače, nacházejícího se na určité vzdálenosti od předmětu, nebo měřením dynamického zrychlení při otáčení akcelerometru. Při tomto způsobu měření však vznikaly velké nepřesnosti, způsobené dlouhou dobou integrace, kdy kvůli šumu senzoru, který v tomto případě již nemohl být dolnoproustním filtrem odstraněn, docházelo k rychlému narůstání chyby, která při druhé integraci rostla kvadratickou závislostí. Jednoduchým a nestandardním řešením by mohlo být použití připravovaného k výrobě levného tříosého kompasu s akcelerometrem LSM303 firmy STM.

## 4 Závěr

V rámci práce byly předvedeny způsoby měření křivek pomocí ohebně a kloubově spojených soustav akcelerometrů. První způsob měření byl předveden na přípravku se třemi měřicími moduly, umístěnými na ohebném rameni. Nehledě na nepřesnosti vyzkoušených aproximačních a interpolačních metod doplnění křivek, lze daný způsob s výhodou využít v aplikacích s menším počtem akcelerometrů, kde měřené křivky netvoří smyčky (například měření křivosti zad, apod.). Při návrhu zařízení pro měření křivek kloubově spojenými senzory nebylo možné najít vhodné klouby, proto daný způsob měření byl předveden pomocí modelu lidské kostry a přípravku, rozmístěvaného na těle.

Při návrhu měřicího systému byly použity akcelerometry MMA7361L, pro snížení jejich nepřesnosti bylo třeba vymyslet způsoby filtrace a víceosé interpretace měřeného signálu. Měřicí moduly se spojují do systému, komunikujícím pomocí sběrnice RS-485. Pro účely přenosu velkého objemu dat, získávaných z něho, bylo použito rozhraní USB v režimu emulace sériového portu, kompatibilním se standardními knihovnamí pro RS-232 komunikaci v Delphi.

Vlastnosti obou metod měření byly předvedeny na modelech křivky, bagru a člověka. Pro tyto účely byl navržen a realizován univerzální postup práce se složitými 3D modely v Delphi.

## 5 Literatura

[1] Dokumentace obvodů: PIC18F2550, PIC18LF1220, MMA7361, MCP1700, SN75LBC176

[2] ROJVANIT, R. AN956: Migrating Applications to USB from RS232 UART with Minimal Impact on PC Software, Application Note Microchip Technology Inc., 2004

Dostupné z WWW: <<http://ww1.microchip.com/downloads/en/AppNotes/00956b.pdf>>.

[3] ADELSBERGER, R, VANUCCI, G, BARNWELL, J, GROSS, M, MATUSIK, W, POPOVIC, J. Practical Motion Capture in Everyday Surroundings. ACM Transactions on Graphics. 26(3) August 2007.

Dostupné z WWW: <<http://portal.acm.org/citation.cfm?id=1276421>>.

## 6 Obsah přiloženého CD

Na přiloženém CD lze najít text diplomové práce, zdrojové kódy PC aplikací, firmware zařízení, USB ovládač a desky plošných spojů. Struktura adresářů je následující:

### **/doc**

Text diplomové práce ve formátech pdf doc.

### **/src/**

Zdrojové kódy, PC ovládače, desky tištěných spojů.

#### **/src/software/**

Zdrojové kódy PC aplikací.

##### **/src/software/krivka**

Zdrojové kódy aplikace křivka.

##### **/src/software/clovek**

Zdrojové kódy aplikace člověk.

#### **/src/firmware/**

Firmware

##### **/src/firmware/merici\_moduly**

Soubor projektu a hex soubor firmware měřících modulů.

##### **/src/firmware/USB\_modul**

Soubor projektu a hex soubor firmware USB modulu.

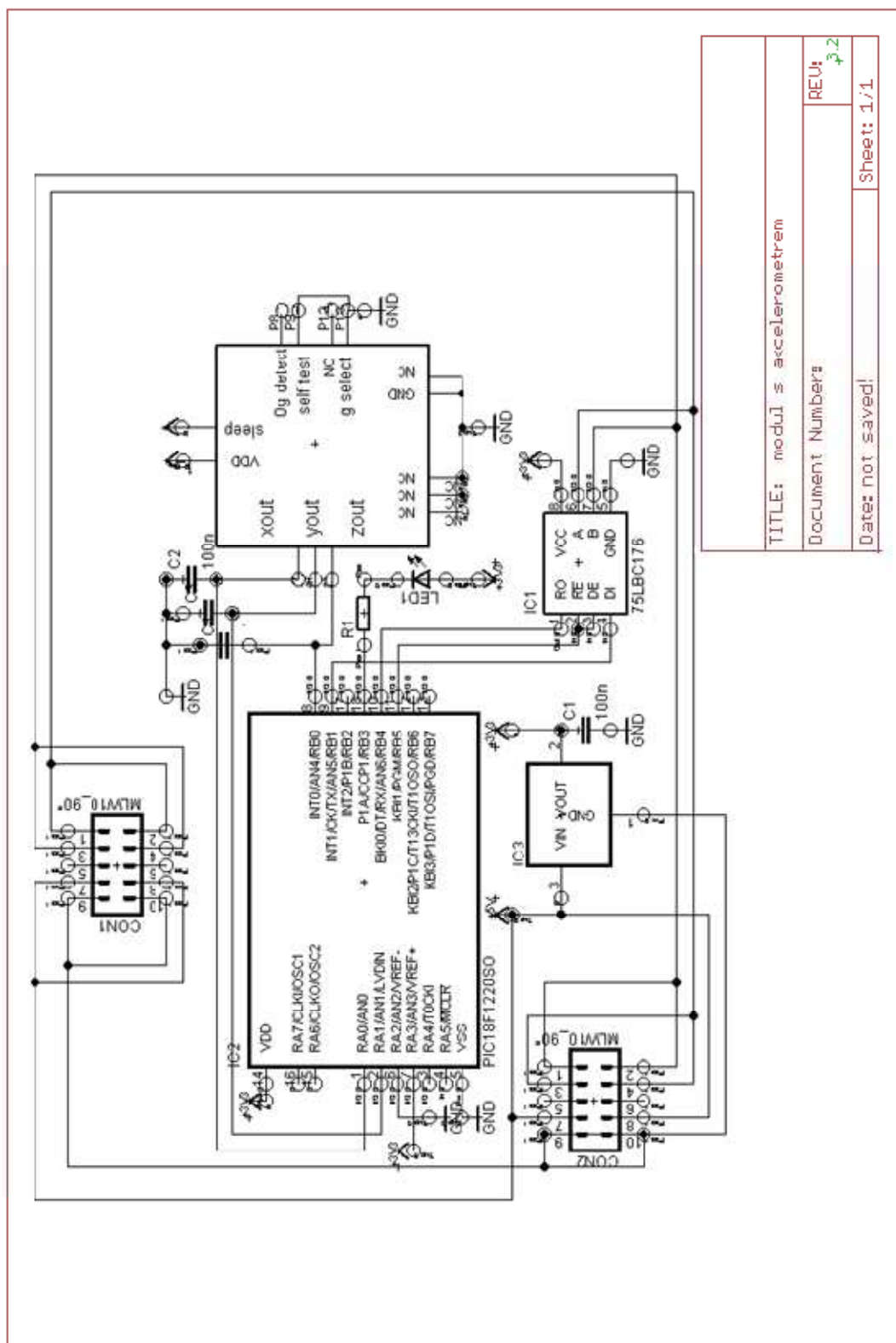
#### **/src/driver**

USB ovládače.

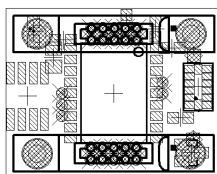
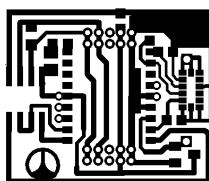
#### **/src/hardware**

Soubory desek plošných spojů

## 7 Přílohy

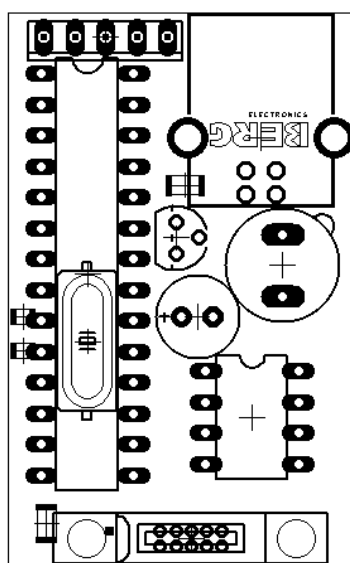
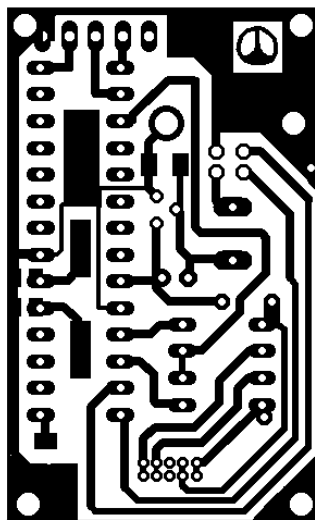


Obr. 30: Schéma zapojení měřících modulů.



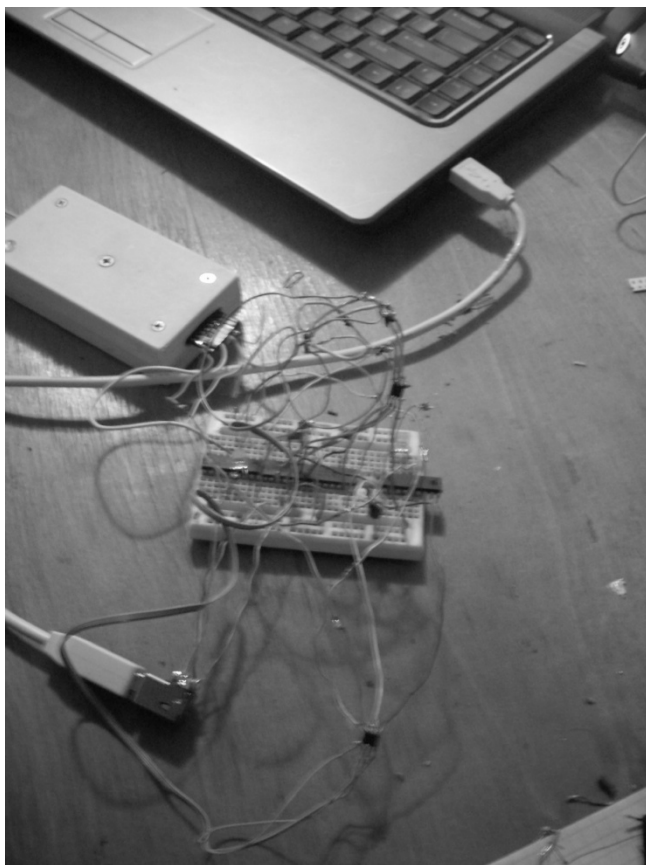
Obr. 31: Deska plošných spojů (Bottom) a rozložení součástek měřicího modulu.





Obr. 33: Deska plošných spojů a rozložení součástek USB modulu.

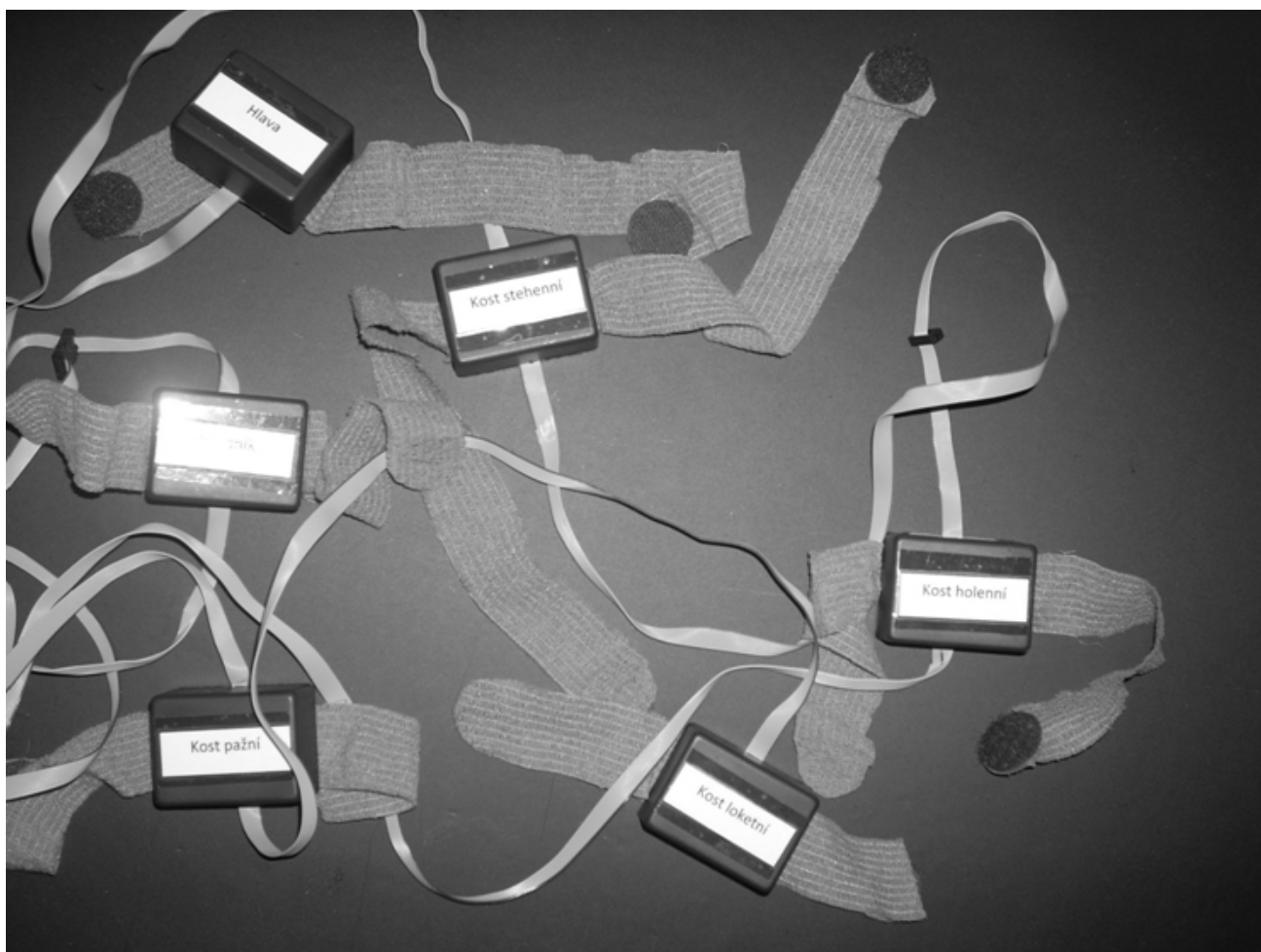




Obr. 34: Pokusné zapojení s akcelerometry MMA7455L.



Obr. 35: Přípravek pro měření ohebně spojenými senzory.



Obr. 36: Přípravek pro měření kloubově spojenými senzory.